

Appendix 5: Income and Assets Variable Creation

Contents

- Household Income and Assets
 - Section 1: Gross Household Income
 - Section 2: Household Net Worth
 - Section 3: Household Poverty Status
 - Section 4: Topcoding Income, Net Worth, and Poverty Ratio
 - Participation in Government Programs
 - Section 1: Unemployment Compensation
 - Section 2: Workers Compensation
 - Section 3: Other Government Assistance
-

Household Income and Assets

Variables Created:

- CV_HH_NET_WORTH_Y
- CV_INCOME_GROSS_YR
- CV_HH_POV_RATIO

Variables Used

Please contact NLS User Services for variable names.

This program creates the household net worth and gross household income variables. The household net worth variable coded as hhworthY is an actual number that results from adding the values of all assets and subtracting liabilities of the household. The gross household income variable coded as groshhIY includes total annual cash receipts before taxes from all sources. The program then creates a ratio comparing the household's total income to federal poverty guidelines based on the number of household residents and the number of members under age 18.

Researchers should note that, like many income and asset variables in the data set, these three variables are topcoded to protect respondent privacy. More information about topcoding is available in the *NLSY97 User's Guide*.

***** SECTION 1: GROSS HOUSEHOLD INCOME *****

/** In income section, I flag three items involving most of the respondents:

I. nonfarm employment income -- flag1400

II. father's income -- flag8700

III. mother's income -- flag9200

flag=0 indicating accurate income reported

flag=1 indicating estimated income reported **/

/*Create the following components of the groshhIY, such as income from wages and salaries, etc. **/

/** Income from non-farm employment (nfarmwgY) **/

nfarmwgY=-4; flag1400=0;

IF (YI_1400=1 OR (YI_1400=-1 AND YI_1600=1) OR (YI_1400=-2 AND YI_1500=1)

OR (YI_1400=-2 AND YI_1500=-1 AND YI_1600=1)) AND (YI_1700~-1 AND YI_1700~-2 AND
YI_1700~-3)

THEN nfarmwgY=YI_1700;

IF (YI_1400=1 OR (YI_1400=-1 AND YI_1600=1) OR (YI_1400=-2 AND YI_1500=1)

OR (YI_1400=-2 AND YI_1500=-1 AND YI_1600=1) OR (YI_1400=-2 AND YI_1500=-3 AND
YI_1600=1)

OR (YI_1400=-3 AND YI_1600=1) OR (YI_1400=-3 AND YI_1500=1))

AND (YI_1700=-1 OR YI_1700=-2 OR YI_1700=-3) then do;

if YI_1800=1 then do; nfarmwgY=2500; flag1400=1; end;

if YI_1800=2 then do; nfarmwgY=7500; flag1400=1; end;

if YI_1800=3 then do; nfarmwgY=17500; flag1400=1; end;

if YI_1800=4 then do; nfarmwgY=37500; flag1400=1; end;

if YI_1800=5 then do; nfarmwgY=75000; flag1400=1; end;

if YI_1800=6 then do; nfarmwgY=175000; flag1400=1; end;

if YI_1800=7 then do; nfarmwgY=250001; flag1400=1; end;

end;

if YI_1400=0 or YI_1500=0 or YI_1600=0 then nfarmwgY=0;

if YI_1600=-1 or YI_1800=-1 then nfarmwgY=-1;

if YI_1500=-2 or YI_1800=-2 then nfarmwgY=-2;

if YI_1600=-3 or YI_1500=-3 or YI_1800=-3 then nfarmwgY=-3;

/** For all the questions below, the youth must be INDEPENDENT (YI_1900=1) **/

/** Income from farm and self-employment (farmwgY) **/

farmwgY=-4;

if YI_2000=1 and YI_2100 ~ -1 and YI_2100 ~ -2 and YI_2100 ~ -3 then farmwgY=YI_2100;

if YI_2000=1 and (YI_2100 eq -1 or YI_2100 eq -2 or YI_2100 eq -3) then do;

if YI_2200=1 then do; farmwgY=-2; end;

if YI_2200=2 then do; farmwgY=2500; end;

if YI_2200=3 then do; farmwgY=7500; end;

if YI_2200=4 then do; farmwgY=17500; end;

if YI_2200=5 then do; farmwgY=37500; end;

if YI_2200=6 then do; farmwgY=75000; end;

if YI_2200=7 then do; farmwgY=175000; end;

if YI_2200=8 then do; farmwgY=250001; end;

end;

if YI_2000=0 then farmwgY=0;

if YI_2000=-1 or YI_2200=-1 then farmwgY=-1;

if YI_2000=-2 or YI_2200=-2 then farmwgY=-2;

if YI_2000=-3 or YI_2200=-3 then farmwgY=-3;

*** Income from worker's comp (wcompY);

```
wcompY=-4;
IF YI_2250=1 THEN DO;
IF YI_2260>=0 THEN wcompY=YI_2260;
IF YI_2260 IN (-1,-2,-3) THEN DO;
IF YI_2270=1 THEN DO;
wcompY=0;
END;
IF YI_2270=2 THEN DO;
wcompY=2500;
END;
IF YI_2270=3 THEN DO;
wcompY=7500;
END;
IF YI_2270=4 THEN DO;
wcompY=17500;
END;
IF YI_2270=5 THEN DO;
wcompY=37500;
END;
IF YI_2270=6 THEN DO;
wcompY=75000;
END;
IF YI_2270=7 THEN DO;
wcompY=175000;
END;
IF YI_2270=8 THEN DO;
wcompY=250001;
END;
END;
END;
END;
IF YI_2250=-1 OR YI_2270=-1 THEN wcompY=-1;
IF YI_2250=-2 OR YI_2270=-2 THEN wcompY=-2;
IF YI_2250=-3 OR YI_2270=-3 THEN wcompY=-3;
```

/** If the youth lives with spouse/partner, non-farm income of the spouse/partner (nfarmwgP) **/

nfarmwgP=-4;

if YI_2300=1 then do;

 if (YI_2400=1 and YI_2600 ~= -1 and YI_2600 ~= -2 and YI_2600 ~= -3) or (YI_2400=-1 and YI_2500=1 and YI_2600 ~= -1 and YI_2600 ~= -2 and YI_2600 ~= -3)

 then nfarmwgP=YI_2600;

 if (YI_2400=1 and (YI_2600 eq -1 or YI_2600 eq -2 or YI_2600 eq -3)) or (YI_2400=-1 and YI_2500=1 and (YI_2600 eq -1 or YI_2600 eq -2 or YI_2600 eq -3)) then do;

 if YI_2700=1 then do; nfarmwgP=2500; end;

 if YI_2700=2 then do; nfarmwgP=7500; end;

 if YI_2700=3 then do; nfarmwgP=17500; end;

 if YI_2700=4 then do; nfarmwgP=37500; end;

 if YI_2700=5 then do; nfarmwgP=75000; end;

```

if YI_2700=6 then do; nfarmwgP=175000; end;
if YI_2700=7 then do; nfarmwgP=250001; end;
end;
end;
if YI_2300=0 or YI_2400=0 or YI_2500=0 then nfarmwgP=0;
if YI_2300=-1 or YI_2500=-1 or YI_2700=-1 then nfarmwgP=-1;
if YI_2300=-2 or YI_2400=-2 or YI_2500=-2 or YI_2700=-2 then nfarmwgP=-2;
if YI_2300=-3 or YI_2400=-3 or YI_2500=-3 or YI_2700=-3 then nfarmwgP=-3;

/** If the youth lives with spouse/partner, farm-related income of the spouse/partner (farmwgP)      **/
farmwgP=-4;
if YI_2300=1 then do;
  if YI_2900=1 and YI_3000 ~= -1 and YI_3000 ~= -2 and YI_3000 ~= -3 then farmwgP=YI_3000;
  if YI_2900=1 and (YI_3000 eq -1 or YI_3000 eq -2 or YI_3000 eq -3) then do;
    if YI_3100=1 then do; farmwgP=-2; end;
    if YI_3100=2 then do; farmwgP=2500; end;
    if YI_3100=3 then do; farmwgP=7500; end;
    if YI_3100=4 then do; farmwgP=17500; end;
    if YI_3100=5 then do; farmwgP=37500; end;
    if YI_3100=6 then do; farmwgP=75000; end;
    if YI_3100=7 then do; farmwgP=175000; end;
    if YI_3100=8 then do; farmwgP=250001; end;
  end;
end;
if YI_2300=0 or YI_2900=0           then farmwgP=0;
if YI_2300=-1 or YI_2900=-1 or YI_3100=-1 then farmwgP=-1;
if YI_2300=-2 or YI_2900=-2 or YI_3100=-2 then farmwgP=-2;
if YI_2300=-3 or YI_2900=-3 or YI_3100=-3 then farmwgP=-3;

```

*** If the youth lives partner/spouse, income received from worker's comp. by the partner/spouse (wcompP);

```

wcompP=-4;
IF YI_2300=1 and YI_3150=1 THEN DO;
IF YI_3160>=0 THEN wcompP=YI_3160;
IF YI_3160 IN (-1,-2,-3) THEN DO;
IF YI_3170=1 THEN DO;
wcompP=500;
END;
IF YI_3170=2 THEN DO;
wcompP=1250;
END;
IF YI_3170=3 THEN DO;
wcompP=3750;
END;
IF YI_3170=4 THEN DO;
wcompP=7500;
END;
IF YI_3170=5 THEN DO;
wcompP=17500;
END;

```

```

IF YI_3170=6 THEN DO;
wcompP=37500;
END;
IF YI_3170=7 THEN DO;
wcompP=50001;
END;
END;
END;
IF YI_2300=-1 OR YI_3150=-1 OR YI_3170=-1 THEN wcompP=-1;
IF YI_2300=-1 OR YI_3150=-2 OR YI_3170=-2 THEN wcompP=-2;
IF YI_2300=-1 OR YI_3150=-3 OR YI_3170=-3 THEN wcompP=-3;

/** Child support payment (childsuY) **/
childsuY=-4;
if YI_3300=1 then do;
  if (YI_3900=1 and YI_4000=1 and (YI_4100 ~= -1 and YI_4100 ~= -2 and YI_4100 ~= -3)) then
    childsuY=YI_4100;
  if (YI_3900=1 and YI_4000=1 and (YI_4100 eq -1 or YI_4100 eq -2 or YI_4100 eq -3)) then do;
    if YI_4200=1 then do; childsuY=500; end;
    if YI_4200=2 then do; childsuY=1750; end;
    if YI_4200=3 then do; childsuY=3750; end;
    if YI_4200=4 then do; childsuY=7500; end;
    if YI_4200=5 then do; childsuY=17500; end;
    if YI_4200=6 then do; childsuY=37500; end;
    if YI_4200=7 then do; childsuY=50001; end;
  end;
end;
if YI_3300=0 or YI_3900=0 or YI_4000=0 then childsuY=0;
if YI_3300=-1 or YI_3900=-1 or YI_4000=-1 or YI_4200=-1 then childsuY=-1;
if YI_3300=-2 or YI_3900=-2 or YI_4000=-2 or YI_4200=-2 then childsuY=-2;
if YI_3300=-1 or YI_3900=-1 or YI_4000=-3 or YI_4100=-3 or YI_4200=-3 then childsuY=-3;

/** Interest payments received by the youth (Y) and his/her partner/spouse. **/
interesY=-4;
if YI_4300=1 then do;
  if (YI_4400 ~= -1 and YI_4400 ~= -2 and YI_4400 ~= -3) then interesY=YI_4400;
  if (YI_4400 eq -1 or YI_4400 eq -2 or YI_4400 eq -3) then do;
    if YI_4500=1 then do; interesY=250; end;
    if YI_4500=2 then do; interesY=750; end;
    if YI_4500=3 then do; interesY=1750; end;
    if YI_4500=4 then do; interesY=3750; end;
    if YI_4500=5 then do; interesY=6250; end;
    if YI_4500=6 then do; interesY=8750; end;
    if YI_4500=7 then do; interesY=10001; end;
  end;
end;
if YI_4300=0 then interesY=0;
if YI_4300=-1 or YI_4500=-1 then interesY=-1;
if YI_4300=-2 or YI_4500=-2 then interesY=-2;
if YI_4300=-3 or YI_4400=-3 or YI_4500=-3 then interesY=-3;

```

```

/** Dividends from stocks and mutual funds. **/
dividend=-4;
if YI_4600=1 then do;
  if (YI_4700 ~= -1 and YI_4700 ~= -2 and YI_4700 ~= -3) then dividend=YI_4700;
  if (YI_4700 eq -1 or YI_4700 eq -2 or YI_4700 eq -3) then do;
    if YI_4800=1 then do; dividend=250; end;
    if YI_4800=2 then do; dividend=750; end;
    if YI_4800=3 then do; dividend=1750; end;
    if YI_4800=4 then do; dividend=3750; end;
    if YI_4800=5 then do; dividend=6250; end;
    if YI_4800=6 then do; dividend=8750; end;
    if YI_4800=7 then do; dividend=10001; end;
  end;
end;
if YI_4600=0 then dividend=0;
if YI_4600=-1 or YI_4800=-1 then dividend=-1;
if YI_4600=-2 or YI_4800=-2 then dividend=-2;
if YI_4600=-3 or YI_4700=-3 or YI_4800=-3 then dividend=-3;

/** Rental income. */
rentalIY=-4;
if YI_4900=1 then do;
  if (YI_5000 ~= -1 and YI_5000 ~= -2 and YI_5000 ~= -3) then rentalIY=YI_5000;
  if (YI_5000 eq -1 or YI_5000 eq -2 or YI_5000 eq -3) then do;
    if YI_5100=1 then do; rentalIY=500; end;
    if YI_5100=2 then do; rentalIY=1750; end;
    if YI_5100=3 then do; rentalIY=3750; end;
    if YI_5100=4 then do; rentalIY=7500; end;
    if YI_5100=5 then do; rentalIY=17500; end;
    if YI_5100=6 then do; rentalIY=37500; end;
    if YI_5100=7 then do; rentalIY=50001; end;
  end;
end;
if YI_4900=0 then rentalIY=0;
if YI_4900=-1 or YI_5100=-1 then rentalIY=-1;
if YI_4900=-2 or YI_5100=-2 then rentalIY=-2;
if YI_4900=-3 or YI_5000=-3 or YI_5100=-3 then rentalIY=-3;

/** Property or money from estates, trusts, annuities or inheritances. */
estatesY=-4;
if YI_5200=1 then do;
  if (YI_5300 ~= -1 and YI_5300 ~= -2 and YI_5300 ~= -3) then estatesY=YI_5300;
  if (YI_5300 eq -1 or YI_5300 eq -2 or YI_5300 eq -3) then do;
    if YI_5400=1 then do; estatesY=2500; end;
    if YI_5400=2 then do; estatesY=7500; end;
    if YI_5400=3 then do; estatesY=17500; end;
    if YI_5400=4 then do; estatesY=37500; end;
    if YI_5400=5 then do; estatesY=75000; end;
    if YI_5400=6 then do; estatesY=175000; end;
    if YI_5400=7 then do; estatesY=250001; end;
  end;
end;

```

```

end;
if YI_5200=0 then estatesY=0;
if YI_5200=-1 or YI_5400=-1 then estatesY=-1;
if YI_5200=-2 or YI_5400=-2 then estatesY=-2;
if YI_5200=-3 or YI_5300=-3 or YI_5400=-3 then estatesY=-3;

/** Income from other sources: SS payments, pension and retirement income, alimony, payments from
insurance policies, etc.      */
pensionY=-4;
if YI_7600=1 then do;
  if (YI_7700 ~= -1 and YI_7700 ~= -2 and YI_7700 ~= -3) then pensionY=YI_7700;
  if (YI_7700 eq -1 or YI_7700 eq -2 or YI_7700 eq -3) then do;
    if YI_7800=1 then do; pensionY=500; end;
    if YI_7800=2 then do; pensionY=1750; end;
    if YI_7800=3 then do; pensionY=3750; end;
    if YI_7800=4 then do; pensionY=7500; end;
    if YI_7800=5 then do; pensionY=17500; end;
    if YI_7800=6 then do; pensionY=37500; end;
    if YI_7800=7 then do; pensionY=50001; end;
  end;
end;
if YI_7600=0 then pensionY=0;
if YI_7600=-1 or YI_7800=-1 then pensionY=-1;
if YI_7600=-2 or YI_7800=-2 then pensionY=-2;
if YI_7600=-3 or YI_7700=-3 or YI_7800=-3 then pensionY=-3;

/** If the youth lived with the father, the father's income.      */
flag8700=0; faincome=-4;
if (HIRELY01=4 or HIRELY02=4 or HIRELY03=4 or HIRELY04=4 or HIRELY05=4 or
HIRELY06=4 or HIRELY07=4 or HIRELY08=4 or HIRELY09=4 or HIRELY10=4 or HIRELY11=4
or HIRELY12=4 or HIRELY13=4 or HIRELY14=4) then do;
  if YI_8700=1 then do;
    if (YI_8800 ~= -1 and YI_8800 ~= -2 and YI_8800 ~= -3) then faincome=YI_8800;
    if (YI_8800 eq -1 or YI_8800 eq -2 or YI_8800 eq -3) then do;
      if YI_8900=1 then do; faincome=2500; flag8700=1; end;
      if YI_8900=2 then do; faincome=7500; flag8700=1; end;
      if YI_8900=3 then do; faincome=17500; flag8700=1; end;
      if YI_8900=4 then do; faincome=37500; flag8700=1; end;
      if YI_8900=5 then do; faincome=75000; flag8700=1; end;
      if YI_8900=6 then do; faincome=175000; flag8700=1; end;
      if YI_8900=7 then do; faincome=250001; flag8700=1; end;
    end;
  end;
  if YI_8700=0 then faincome=0;
  if YI_8700=-1 or YI_8900=-1 then faincome=-1;
  if YI_8700=-2 or YI_8900=-2 then faincome=-2;
  if YI_8700=-3 or YI_8800=-3 or YI_8900=-3 then faincome=-3;
end;

/** If the youth lived with the mother, the mother's income.      */
flag9200=0;

```

```

maincome=-4;
if (HIRELY01=3 or HIRELY02=3 or HIRELY03=3 or HIRELY04=3 or HIRELY05=3 or
HIRELY06=3 or HIRELY07=3 or HIRELY08=3 or HIRELY09=3 or HIRELY10=3 or HIRELY11=3
or HIRELY12=3 or HIRELY13=3 or HIRELY14=3) then do;
  if YI_9200=1 then do;
    if (YI_9300 ~= -1 and YI_9300 ~= -2 and YI_9300 ~= -3) then maincome=YI_9300;
    if (YI_9300 eq -1 or YI_9300 eq -2 or YI_9300 eq -3) then do;
      if YI_9400=1 then do; maincome=2500; flag9200=1; end;
      if YI_9400=2 then do; maincome=7500; flag9200=1; end;
      if YI_9400=3 then do; maincome=17500; flag9200=1; end;
      if YI_9400=4 then do; maincome=37500; flag9200=1; end;
      if YI_9400=5 then do; maincome=75000; flag9200=1; end;
      if YI_9400=6 then do; maincome=175000; flag9200=1; end;
      if YI_9400=7 then do; maincome=250001; flag9200=1; end;
    end;
  end;
  if YI_9200=0 then maincome=0;
  if YI_9200=-1 or YI_9400=-1 then maincome=-1;
  if YI_9200=-2 or YI_9400=-2 then maincome=-2;
  if YI_9200=-3 or YI_9400=-3 then maincome=-3;
end;

```

```

/** If the youth lives with male guardian, the male guardian's income. **/
mgincome=-4;
if YI_9600=1 then do;
  if (YI_9700=1 and (YI_9800 ~= -1 and YI_9800 ~= -2 and YI_9800 ~= -3)) then
mgincome=YI_9800;
  if (YI_9700=1 and (YI_9800 eq -1 or YI_9800 eq -2 or YI_9800 eq -3)) then do;
    if YI_9900=1 then do; mgincome=2500; end;
    if YI_9900=2 then do; mgincome=7500; end;
    if YI_9900=3 then do; mgincome=17500; end;
    if YI_9900=4 then do; mgincome=37500; end;
    if YI_9900=5 then do; mgincome=75000; end;
    if YI_9900=6 then do; mgincome=175000; end;
    if YI_9900=7 then do; mgincome=250001; end;
  end;
end;
if YI_9700=0 then mgincome=0;
if YI_9600=-1 or YI_9700=-1 or YI_9900=-1 then mgincome=-1;
if YI_9600=-2 or YI_9700=-2 or YI_9900=-2 then mgincome=-2;
if YI_9600=-3 or YI_9700=-3 or YI_9800=-3 or YI_9900=-3 then mgincome=-3;

```

```

/** If the youth lives with female guardian, the female guardian's income. **/
fgincome=-4;
if YI_10100=1 then do;
  if (YI_10200=1 and (YI_10300 ~= -1 and YI_10300 ~= -2 and YI_10300 ~= -3)) then
fgincome=YI_10300;
  if (YI_10100=1 and YI_10200=1 and (YI_10300 eq -1 or YI_10300 eq -2 or YI_10300 eq -3)) then
do;
  if YI_10400=1 then do; fgincome=2500; end;
  if YI_10400=2 then do; fgincome=7500; end;

```

```

if YI_10400=3 then do; fgincome=17500; end;
if YI_10400=4 then do; fgincome=37500; end;
if YI_10400=5 then do; fgincome=75000; end;
if YI_10400=6 then do; fgincome=175000; end;
if YI_10400=7 then do; fgincome=250001; end;
end;
end;
if YI_10200=0 then fgincome=0;
if YI_10100=-1 or YI_10200=-1 or YI_10400=-1 then fgincome=-1;
if YI_10100=-2 or YI_10200=-2 or YI_10400=-2 then fgincome=-2;
if YI_10100=-3 or YI_10200=-3 or YI_10300=-3 or YI_10400=-3 then fgincome=-3;

/** Income from household members of 14 or older other than spouse/partner. **/
array otfamI      (14) otfamI01-otfamI14;
array YI_11100 (14) YI_1110001-YI_1110014;
array YI_11600 (14) YI_1160001-YI_1160014;
array YI_11700 (14) YI_1170001-YI_1170014;

do I=1 to DIM(otfamI);
  otfamI(I)=-4;
  if YI_10800=1 then do;
    if YI_11100(I)=0 and YI_11600(I) ~= -1 and YI_11600(I) ~= -2 and YI_11600(I) ~= -3 and
YI_11600(I) ~= -4 then otfamI(I)=YI_11600(I);
    if YI_11100(I)=0 and (YI_11600(I) eq -1 or YI_11600(I) eq -2 or YI_11600(I) eq -3) then do;
      if YI_11700(I)=1 then do; otfamI(I)=2500; end;
      if YI_11700(I)=2 then do; otfamI(I)=7500; end;
      if YI_11700(I)=3 then do; otfamI(I)=17500; end;
      if YI_11700(I)=4 then do; otfamI(I)=37500; end;
      if YI_11700(I)=5 then do; otfamI(I)=75000; end;
      if YI_11700(I)=6 then do; otfamI(I)=175000; end;
      if YI_11700(I)=7 then do; otfamI(I)=250001; end;
    end;
    if YI_11100(I)=-1 or YI_11700(I)=-1 then otfamI(I)=-1;
    if YI_11100(I)=-2 or YI_11700(I)=-2 then otfamI(I)=-2;
    if YI_11100(I)=-3 or YI_11600(I)=-3 or YI_11700(I)=-3 then otfamI(I)=-3;
  end;
end;
end;

/** We now create ggroshhIY;      **/
groshhIY=0;
if YI_1900=1 then do;
  if nfarmwgY NOT in (-1,-2,-3,-4) then groshhIY=groshhIY+nfarmwgY;
  if farmwgY NOT in (-1,-2,-3,-4) then groshhIY=groshhIY+farmwgY;
  if nfarmwgP NOT in (-1,-2,-3,-4) then groshhIY=groshhIY+nfarmwgP;
  if farmwgP NOT in (-1,-2,-3,-4) then groshhIY=groshhIY+farmwgP;
  if childsuY NOT in (-1,-2,-3,-4) then groshhIY=groshhIY+childsuY;
  if interesY NOT in (-1,-2,-3,-4) then groshhIY=groshhIY+interesY;
  if dividend NOT in (-1,-2,-3,-4) then groshhIY=groshhIY+dividend;
  if rentalIY NOT in (-1,-2,-3,-4) then groshhIY=groshhIY+rentalIY;
  if estatesY NOT in (-1,-2,-3,-4) then groshhIY=groshhIY+estatesY;
  if pensionY NOT in (-1,-2,-3,-4) then groshhIY=groshhIY+pensionY;

```

```

if faincome NOT in (-1,-2,-3,-4) then groshhIY=groshhIY+faincome;
if maincome NOT in (-1,-2,-3,-4) then groshhIY=groshhIY+maincome;
if mgincome NOT in (-1,-2,-3,-4) then groshhIY=groshhIY+mgincome;
if fgincome NOT in (-1,-2,-3,-4) then groshhIY=groshhIY+fgincome;
if wcompY NOT IN (-1,-2,-3,-4) then groshhIY=groshhIY+wcompY;
if wcompP NOT IN (-1,-2,-3,-4) then groshhIY=groshhIY+wcompP;
if prgamt NOT in (-1,-2,-3,-4) then groshhIY=groshhIY+prgamt;
do I=1 to DIM(otfamI);
  if otfamI[I] NOT in (-1,-2,-3,-4) then groshhIY=groshhIY+otfamI[I];
end;
end;

```

IF YI_1900=-1 OR YI_8500=-1
OR (nfarmwgY=-1 OR farmwgY=-1 OR nfarmwgP=-1 OR farmwgP=-1 OR childsuY=-1
OR interesY=-1 OR dividend=-1 OR rentalIY=-1 OR estatesY=-1 OR pensionY=-1
OR faincome=-1 OR maincome=-1 OR mgincome=-1 OR fgincome=-1 OR wcompY=-1
OR wcompP=-1 OR prgamt=-1
OR otfamI01=-1 OR otfamI02=-1 OR otfamI03=-1 OR otfamI04=-1 OR otfamI05=-1
OR otfamI06=-1 OR otfamI07=-1 OR otfamI08=-1 OR otfamI09=-1 OR otfamI10=-1
OR otfamI11=-1 OR otfamI12=-1 OR otfamI13=-1 OR otfamI14=-1) THEN groshhIY=-1;

IF YI_1900=-2 OR YI_8500=-2
OR (nfarmwgY=-2 OR farmwgY=-2 OR nfarmwgP=-2 OR farmwgP=-2 OR childsuY=-2
OR interesY=-2 OR dividend=-2 OR rentalIY=-2 OR pensionY=-2 OR estatesY=-2
OR faincome=-2 OR maincome=-2 OR mgincome=-2 OR fgincome=-2 OR wcompY=-2
OR wcompP=-2 OR prgamt=-2
OR otfamI01=-2 OR otfamI02=-2 OR otfamI03=-2 OR otfamI04=-2 OR otfamI05=-2
OR otfamI06=-2 OR otfamI07=-2 OR otfamI08=-2 OR otfamI09=-2 OR otfamI10=-2
OR otfamI11=-2 OR otfamI12=-2 OR otfamI13=-2 OR otfamI14=-2) THEN groshhIY=-2;

IF YI_1900=-3 OR YI_8500=-3
OR (nfarmwgY=-3 OR farmwgY=-3 OR nfarmwgP=-3 OR farmwgP=-3 OR childsuY=-3
OR interesY=-3 OR dividend=-3 OR rentalIY=-3 OR pensionY=-3 OR estatesY=-3
OR faincome=-3 OR maincome=-3 OR mgincome=-3 OR fgincome=-3 OR wcompY=-3
OR wcompP=-3 OR prgamt=-3
OR otfamI01=-3 OR otfamI02=-3 OR otfamI03=-3 OR otfamI04=-3 OR otfamI05=-3
OR otfamI06=-3 OR otfamI07=-3 OR otfamI08=-3 OR otfamI09=-3 OR otfamI10=-3
OR otfamI11=-3 OR otfamI12=-3 OR otfamI13=-3 OR otfamI14=-3) THEN groshhIY=-3;

if YI_1900=0 or YI_1900=-4 then groshhIY=-4;

/** give the people who were not interviewed in round 6 value -5. **/
if YAS050=-5 then groshhIY=-5;

/***** SECTION 2: HOUSEHOLD NET WORTH *****/

/** In asset section, I flag top three items involved the most respondents:

- I. savings -- flag4400
- II. present value of vehicles owned -- flag4790 III. money still owed on these vehicles -- flag4840

flag=0 indicating accurate income reported flag=1 indicating estimated income reported. **/

ARRAY VARSL (101) YAS1864L YAS2144L YAS2174L YAS2554L YAS2764L
YAS3014L YAS3384L YAS3754L YAS3814L YAS3884L
YAS3924L YAS4034L YAS4164L YAS4294L YAS4424L
YAS4554L YAS4684L YAS4814L YAS4844L YAS4904L YAS5224L
YAS5084L01-YAS5084L20 YAS5084L01-YAS5084L20 YAS5174L01-YAS5174L20 YAS5174L01-YAS5174L20;

ARRAY VARSU (101) YAS1864U YAS2144U YAS2174U YAS2554U YAS2764U
YAS3014U YAS3384U YAS3754U YAS3814U YAS3884U
YAS3924U YAS4034U YAS4164U YAS4294U YAS4424U
YAS4554U YAS4684U YAS4814U YAS4844U YAS4904U YAS5224U
YAS5084U01-YAS5084U20 YAS5084U01-YAS5084U20 YAS5174U01-YAS5174U20
YAS5174U01-YAS5174U20;

DO I=1 TO DIM(VARSL);
IF VARSL(I)>=0 AND VARSU(I)>=0 AND VARSL(I)>VARSU(I) THEN FLAG_RANGE=1;
END;

*** First we create the following components of the hhworthY such as present value of ranch etc.;

/** If the youth OWNS some land, ITS PRESENT VALUE. **/
pvranch=0;
if YAS1400=1 then do;
/** If the youth reported owning all of it. **/
 if YAS1610=1 or YAS1610=2 then do;
 if YAS1860=1 then pvranch=YAS1862;
 if YAS1860=2 then pvranch=YAS1864L+(YAS1864U-YAS1864L)/2;
 if ((YAS1860 ~= 1 and YAS1860 ~= 2) or (YAS1862=-1 or YAS1862=-2)) then do;
 if YAS1866=1 then do; pvranch=12500; end;
 if YAS1866=2 then do; pvranch=37500; end;
 if YAS1866=3 then do; pvranch=75000; end;
 if YAS1866=4 then do; pvranch=175000; end;
 if YAS1866=5 then do; pvranch=375000; end;
 if YAS1866=6 then do; pvranch=750000; end;
 if YAS1866=7 then do; pvranch=1000001; end;
 end;
 end;
/** If the youth reported owning only part of it **/
 if (YAS1610=3 or YAS1610=4) and YAS2120=100 then do;
 if YAS2140=1 then pvranch=YAS2142;
 if YAS2140=2 then pvranch=(YAS2144L+(YAS2144U-YAS2144L)/2);
 if ((YAS2140 ~= 1 and YAS2140 ~= 2) or (YAS2142=-1 or YAS2142=-2)) then do;
 if YAS2146=1 then do; pvranch=12500; end;
 if YAS2146=2 then do; pvranch=37500; end;
 if YAS2146=3 then do; pvranch=75000; end;
 if YAS2146=4 then do; pvranch=175000; end;
 if YAS2146=5 then do; pvranch=375000; end;
 if YAS2146=6 then do; pvranch=750000; end;
 if YAS2146=7 then do; pvranch=1000001; end;

```
    end;  
end;
```

*** If the youth reported owning only part of it (NE 100%);

```
if (YAS1610=3 or YAS1610=4) and YAS2120 ~= 100 then do;  
    if YAS2170=1 then pvranch=YAS2172;  
    if YAS2170=2 then pvranch=(YAS2174L+(YAS2174U-YAS2174L)/2);  
    if ((YAS2170 ~= 1 and YAS2170 ~= 2) or (YAS2172=-1 or YAS2172=-2)) then do;  
        if YAS2176=1 then do; pvranch=12500; end;  
        if YAS2176=2 then do; pvranch=37500; end;  
        if YAS2176=3 then do; pvranch=75000; end;  
        if YAS2176=4 then do; pvranch=175000; end;  
        if YAS2176=5 then do; pvranch=375000; end;  
        if YAS2176=6 then do; pvranch=750000; end;  
        if YAS2176=7 then do; pvranch=1000001; end;
```

```
    end;
```

```
end;
```

```
end;
```

```
IF YAS1400=-1 OR YAS1610=-1 OR YAS1864L=-1 OR YAS1864U=-1 OR YAS1866=-1 OR  
YAS2144L=-1 OR YAS2144U=-1  
OR YAS2146=-1 OR YAS2174L=-1 OR YAS2174U=-1 OR YAS2176=-1 THEN pvranch=-1;  
IF YAS1400=-2 OR YAS1610=-2 OR YAS1864L=-2 OR YAS1864U=-2 OR YAS1866=-2 OR  
YAS2144L=-2 OR YAS2144U=-2  
OR YAS2146=-2 OR YAS2174L=-2 OR YAS2174U=-2 OR YAS2176=-2 THEN pvranch=-2;  
IF YAS1400=-3 OR YAS1610=-3 OR YAS1864L=-3 OR YAS1864U=-3 OR YAS1866=-3 OR  
YAS2144L=-3 OR YAS2144U=-3  
OR YAS2146=-3 OR YAS2174L=-3 OR YAS2174U=-3 OR YAS2176=-3 THEN pvranch=-3;
```

/** If the youth owns a mobile home and its site, its PRESENT VALUE. **/

```
pvmbst=0;
```

```
if YAS1500=1 then do;
```

```
    if (YAS2520=1 or YAS2520=2) and YAS2550=1 then pvmbst=YAS2552;  
    if (YAS2520=1 or YAS2520=2) and YAS2550=2 then pvmbst=YAS2554L+(YAS2554U-  
YAS2554L)/2;  
    if (YAS2520=1 or YAS2520=2) and ((YAS2550 ~= 1 and YAS2550 ~= 2) or (YAS2552=-1 or  
YAS2552=-2)) then do;  
        if YAS2556=1 then do; pvmbst=500; end;  
        if YAS2556=2 then do; pvmbst=1750; end;  
        if YAS2556=3 then do; pvmbst=3750; end;  
        if YAS2556=4 then do; pvmbst=7500; end;  
        if YAS2556=5 then do; pvmbst=17500; end;  
        if YAS2556=6 then do; pvmbst=37500; end;  
        if YAS2556=7 then do; pvmbst=50001; end;
```

```
    end;
```

```
end;
```

```
if YAS1500=-1 or YAS2520=-1 or YAS2554L=-1 or YAS2554U=-1 or YAS2556=-1 then pvmbst=-1;  
if YAS1500=-2 or YAS2520=-2 or YAS2554L=-2 or YAS2554U=-2 or YAS2556=-2 then pvmbst=-2;  
if YAS1500=-3 or YAS2520=-3 or YAS2552=-3 or YAS2554L=-3 or YAS2554U=-3 or YAS2556=-3  
then pvmbst=-3;
```

```

/** If the youth owns only the mobile home, its PRESENT VALUE. **/
pvmb=0;
if YAS1500=1 then do;
  if (YAS2520=3 or YAS2520=4) and YAS2760=1 then pvmb=YAS2762;
  if (YAS2520=3 or YAS2520=4) and YAS2760=2 then pvmb=(YAS2764L+(YAS2764U-
YAS2764L)/2);
  if (YAS2520=3 or YAS2520=4) and ((YAS2760 ~= 1 and YAS2760 ~= 2) or (YAS2762=-1 or
YAS2762=-2)) then do;
    if YAS2766=1 then do; pvmb=12500; end;
    if YAS2766=2 then do; pvmb=37500; end;
    if YAS2766=3 then do; pvmb=75000; end;
    if YAS2766=4 then do; pvmb=175000; end;
    if YAS2766=5 then do; pvmb=375000; end;
    if YAS2766=6 then do; pvmb=750000; end;
    if YAS2766=7 then do; pvmb=1000001; end;
  end;
end;
if YAS1500=-1 or YAS2520=-1 or YAS2764L=-1 or YAS2764U=-1 or YAS2766=-1 then pvmb=-1;
if YAS1500=-2 or YAS2520=-2 or YAS2764L=-2 or YAS2764U=-2 or YAS2766=-2 then pvmb=-2;
if YAS1500=-3 or YAS2520=-3 or YAS2762=-3 or YAS2764L=-3 or YAS2764U=-3 or YAS2766=-3
then pvmb=-3;

```

```

/** If the youth owns only a mobile home site, ITS PRESENT VALUE. **/
pvst=0;
if YAS1500=1 then do;
  if (YAS2520=5 or YAS2520=6) and YAS3010=1 then pvst=YAS3012;
  if (YAS2520=5 or YAS2520=6) and YAS3010=2 then pvst=(YAS3014L+(YAS3014U-
YAS3014L)/2);
  if (YAS2520=5 or YAS2520=6) and ((YAS3010 ~= 1 and YAS3010 ~= 2) or (YAS3012=-1 or
YAS3012=-2)) then do;
    if YAS3016=1 then do; pvst=12500; end;
    if YAS3016=2 then do; pvst=57500; end;
    if YAS3016=3 then do; pvst=75000; end;
    if YAS3016=4 then do; pvst=175000; end;
    if YAS3016=5 then do; pvst=575000; end;
    if YAS3016=6 then do; pvst=750000; end;
    if YAS3016=7 then do; pvst=1000001; end;
  end;
end;
if YAS2520=-1 or YAS3014L=-1 or YAS3014U=-1 or YAS3016=-1 or YAS1500=-1 then pvst=-1;
if YAS2520=-2 or YAS3014L=-2 or YAS3014U=-2 or YAS3016=-2 or YAS1500=-2 then pvst=-2;
if YAS2520=-3 or YAS3012=-3 or YAS3014L=-3 or YAS3014U=-3 or YAS3016=-3 or YAS1500=-3
then pvst=-3;

```

```

/** If the youth owns an apartment, its PRESENT VALUE; **/
pvapthm=0;
if YAS3310=1 or YAS3310=2 then do;
  if YAS3380=1 then pvapthm=YAS3382;
  if YAS3380=2 then pvapthm=(YAS3384L+(YAS3384U-YAS3384L)/2);
  if ((YAS3380 ~= 1 and YAS3380 ~= 2) or (YAS3382=-1 or YAS3382=-2)) then do;
    if YAS3386=1 then do; pvapthm=500; end;

```

```

if YAS3386=2 then do; pvapthm=1750; end;
if YAS3386=3 then do; pvapthm=3750; end;
if YAS3386=4 then do; pvapthm=7500; end;
if YAS3386=5 then do; pvapthm=17500; end;
if YAS3386=6 then do; pvapthm=37500; end;
if YAS3386=7 then do; pvapthm=50001; end;
end;
end;
if YAS3310=-1 or YAS3384L=-1 or YAS3384U=-1 or YAS3386=-1 then pvapthm=-1;
if YAS3310=-2 or YAS3384L=-2 or YAS3384U=-2 or YAS3386=-2 then pvapthm=-2;
if YAS3310=-3 or YAS3382=-3 or YAS3384L=-3 or YAS3384U=-3 or YAS3386=-3 then pvapthm=-3;

/** If there is mortgage or land contract on youth's land or property;      */
mortgagY=0;
if YAS3740=1 or YAS3740=2 then do;
  if YAS3750=1 and YAS3752>=0 then mortgagY=YAS3752;
  if YAS3750=2 and YAS3754U>=0 and YAS3754L>=0 then mortgagY=(YAS3754L+(YAS3754U-
YAS3754L)/2);
  if ((YAS3750 ~= 1 and YAS3750 ~= 2) or (YAS3752=-1 or YAS3752=-2)) then do;
    if YAS3756=1 then do; mortgagY=500; end;
    if YAS3756=2 then do; mortgagY=1750; end;
    if YAS3756=3 then do; mortgagY=3750; end;
    if YAS3756=4 then do; mortgagY=7500; end;
    if YAS3756=5 then do; mortgagY=17500; end;
    if YAS3756=6 then do; mortgagY=37500; end;
    if YAS3756=7 then do; mortgagY=50001; end;
  end;
end;
if YAS3740=-1 or YAS3754L=-1 or YAS3754U=-1 or YAS3756=-1 then mortgagY=-1;
if YAS3740=-2 or YAS3754L=-2 or YAS3754U=-2 or YAS3756=-2 then mortgagY=-2;
if YAS3740=-3 or YAS3752=-3 or YAS3754L=-3 or YAS3754U=-3 or YAS3756=-3 then mortgagY=-3;

/** If youth took loans from relatives or friends when bought or remodeled the residence. */
loanowed=0;
if YAS3790=1 then do;
  if YAS3810=1 and YAS3812>=0 then loanowed=YAS3812;
  if YAS3810=2 and YAS3814U>=0 and YAS3814L>=0 then loanowed=(YAS3814L+(YAS3814U-
YAS3814L)/2);
  if ((YAS3810 ~= 1 and YAS3810 ~= 2) or (YAS3812=-1 or YAS3812=-2)) then do;
    if YAS3816=1 then do; loanowed=12500; end;
    if YAS3816=2 then do; loanowed=37500; end;
    if YAS3816=3 then do; loanowed=75000; end;
    if YAS3816=4 then do; loanowed=175000; end;
    if YAS3816=5 then do; loanowed=375000; end;
    if YAS3816=6 then do; loanowed=750000; end;
    if YAS3816=7 then do; loanowed=1000001; end;
  end;
end;
if YAS3810=-1 or YAS3790=-1 or YAS3814L=-1 or YAS3814U=-1 or YAS3816=-1 then loanowed=-1;

```

if YAS3810=-2 or YAS3790=-2 or YAS3814L=-2 or YAS3814U=-2 or YAS3816=-2 then loanowed=-2;
 if YAS3810=-3 or YAS3790=-3 or YAS3812=-3 or YAS3814L=-3 or YAS3814U=-3 or YAS3816=-3 then loanowed=-3;

/** If youth has unpaid loans from round 5, total amount owed. by youth **/
 stilowed=0;
 if (YAS3840=1 and YAS3850=1) or YAS3860=1 then do;
 if YAS3880=1 and YAS3882>=0 then stilowed=YAS3882;
 if YAS3880=2 and YAS3884U>=0 and YAS3884L>=0 then stilowed=(YAS3884L+(YAS3884U-YAS3884L)/2);
 if ((YAS3880 ~= 1 and YAS3880 ~= 2) or (YAS3882=-1 or YAS3882=-2)) then do;
 if YAS3886=1 then do; stilowed=500; end;
 if YAS3886=2 then do; stilowed=1750; end;
 if YAS3886=3 then do; stilowed=3750; end;
 if YAS3886=4 then do; stilowed=7500; end;
 if YAS3886=5 then do; stilowed=17500; end;
 if YAS3886=6 then do; stilowed=37500; end;
 if YAS3886=7 then do; stilowed=50001; end;
 end;
 end;
 if YAS3860=-1 or YAS3884L=-1 or YAS3884U=-1 or YAS3886=-1 then stilowed=-1;
 if YAS3860=-2 or YAS3884L=-2 or YAS3884U=-2 or YAS3886=-2 then stilowed=-2;
 if YAS3860=-3 or YAS3882=-3 or YAS3884L=-3 or YAS3884U=-3 or YAS3886=-3 then stilowed=-3;

/** If youth has second mortgages **/
 secmortY=0;
 if YAS3910=1 then do;
 if YAS3920=1 and YAS3922>=0 then secmortY=YAS3922;
 if YAS3920=2 and YAS3924U>=0 and YAS3924L>=0 then secmortY=(YAS3924L+(YAS3924U-YAS3924L)/2);
 if ((YAS3920 ~= 1 and YAS3920 ~= 2) or (YAS3922=-1 or YAS3922=-2)) then do;
 if YAS3926=1 then do; secmortY=500; end;
 if YAS3926=2 then do; secmortY=1750; end;
 if YAS3926=3 then do; secmortY=3750; end;
 if YAS3926=4 then do; secmortY=7500; end;
 if YAS3926=5 then do; secmortY=17500; end;
 if YAS3926=6 then do; secmortY=37500; end;
 if YAS3926=7 then do; secmortY=50001; end;
 end;
 end;
 if YAS3910=-1 or YAS3924L=-1 or YAS3924U=-1 or YAS3926=-1 then secmortY=-1;
 if YAS3910=-2 or YAS3924L=-2 or YAS3924U=-2 or YAS3926=-2 then secmortY=-2;
 if YAS3910=-3 or YAS3922=-3 or YAS3924L=-3 or YAS3924U=-3 or YAS3926=-3 then secmortY=-3;

/** If youth still has to pay taxes on the property. **/
 proptaxY=0;
 if YAS3950=1 then do;
 if YAS3960=1 and YAS3962>=0 then proptaxY=YAS3962;
 if YAS3960=2 and YAS3964U>=0 and YAS3964L>=0 then proptaxY=(YAS3964L+(YAS3964U-

```

YAS3964L)/2);
if ((YAS3960 ~= 1 and YAS3960 ~= 2) or (YAS3962=-1 or YAS3962=-2)) then do;
  if YAS3966=1 then do; proptaxY=500; end;
  if YAS3966=2 then do; proptaxY=1750; end;
  if YAS3966=3 then do; proptaxY=3750; end;
  if YAS3966=4 then do; proptaxY=7500; end;
  if YAS3966=5 then do; proptaxY=17500; end;
  if YAS3966=6 then do; proptaxY=37500; end;
  if YAS3966=7 then do; proptaxY=50001; end;
end;
end;
if YAS3950=-1 or YAS3964L=-1 or YAS3964U=-1 or YAS3966=-1 then proptaxY=-1;
if YAS3950=-2 or YAS3964L=-2 or YAS3964U=-2 or YAS3966=-2 then proptaxY=-2;
if YAS3950=-3 or YAS3962=-3 or YAS3964L=-3 or YAS3964U=-3 or YAS3966=-3 then proptaxY=-3;

/** If youth owns a business, partnership or professional practice, its PRESENT VALUE.      */
pvbuss=0;
if YAS4010=1 or YAS4010=2 then do;
  if YAS4030=1 then pvbuss=YAS4032;
  if YAS4030=2 then pvbuss=(YAS4034L+(YAS4034U-YAS4034L)/2);
  if ((YAS4030 ~= 1 and YAS4030 ~= 2) or (YAS4032=-1 or YAS4032=-2)) then do;
    if YAS4036=1 then do; pvbuss=12500; end;
    if YAS4036=2 then do; pvbuss=37500; end;
    if YAS4036=3 then do; pvbuss=75000; end;
    if YAS4036=4 then do; pvbuss=175000; end;
    if YAS4036=5 then do; pvbuss=375000; end;
    if YAS4036=6 then do; pvbuss=750000; end;
    if YAS4036=7 then do; pvbuss=1000001; end;
  end;
end;
if YAS4010=-1 or YAS4034L=-1 or YAS4034U=-1 or YAS4036=-1 then pvbuss=-1;
if YAS4010=-2 or YAS4034L=-2 or YAS4034U=-2 or YAS4036=-2 then pvbuss=-2;
if YAS4010=-3 or YAS4032=-3 or YAS4034L=-3 or YAS4034U=-3 or YAS4036=-3 then pvbuss=-3;

/** If youth owns a second real estate, its PRESENT VALUE.      */
secrestY=0;
if YAS4140=1 or YAS4140=2 then do;
  if YAS4160=1 then secrestY=YAS4162;
  if YAS4160=2 then secrestY=(YAS4164L+(YAS4164U-YAS4164L)/2);
  if ((YAS4160 ~= 1 and YAS4160 ~= 2) or (YAS4162=-1 or YAS4162=-2)) then do;
    if YAS4166=1 then do; secrestY=12500; end;
    if YAS4166=2 then do; secrestY=37500; end;
    if YAS4166=3 then do; secrestY=75000; end;
    if YAS4166=4 then do; secrestY=175000; end;
    if YAS4166=5 then do; secrestY=375000; end;
    if YAS4166=6 then do; secrestY=750000; end;
    if YAS4166=7 then do; secrestY=1000001; end;
  end;
end;
if YAS4140=-1 or YAS4164L=-1 or YAS4164U=-1 or YAS4166=-1 then secrestY=-1;

```

```

if YAS4140=-2 or YAS4164L=-2 or YAS4164U=-2 or YAS4166=-2 then secrestY=-2;
if YAS4140=-3 or YAS4162=-3 or YAS4164L=-3 or YAS4164U=-3 or YAS4166=-3 then secrestY=-3;

/* If youth has retirement plans or pensions */
retireY=0;
if YAS4270=1 or YAS4270=2 or YAS4270=3 then do;
  if YAS4290=1 and YAS4292>=0 then retireY=YAS4292;
  if YAS4290=2 and YAS4294U>=0 and YAS4294L>=0 then retireY=(YAS4294L+(YAS4294U-
YAS4294L)/2);
  if ((YAS4290 ~= 1 and YAS4290 ~= 2) or (YAS4292=-1 or YAS4292=-2)) then do;
    if YAS4296=1 then do; retireY=2500; end;
    if YAS4296=2 then do; retireY=7500; end;
    if YAS4296=3 then do; retireY=17500; end;
    if YAS4296=4 then do; retireY=37500; end;
    if YAS4296=5 then do; retireY=75000; end;
    if YAS4296=6 then do; retireY=175000; end;
    if YAS4296=7 then do; retireY=250001; end;
  end;
end;
if YAS4270=-1 or YAS4294L=-1 or YAS4294U=-1 or YAS4296=-1 then retireY=-1;
if YAS4270=-2 or YAS4294L=-2 or YAS4294U=-2 or YAS4296=-2 then retireY=-2;
if YAS4270=-3 or YAS4292=-3 or YAS4294L=-3 or YAS4294U=-3 or YAS4296=-3 then retireY=-3;

/** If youth has savings in saving accounts, money market, funds, trusts,; */
savingsY=0; flag4400=0;
if YAS4400=1 or YAS4400=2 or YAS4400=3 then do;
  if YAS4420=1 and YAS4422>=0 then savingsY=YAS4422;
  if YAS4420=2 and YAS4424U>=0 and YAS4424L>=0 then savingsY=(YAS4424L+(YAS4424U-
YAS4424L)/2);
  if ((YAS4420 ~= 1 and YAS4420 ~= 2) or (YAS4422=-1 or YAS4422=-2)) then do;
    if YAS4426=1 then do; savingsY=500; flag4400=1; end;
    if YAS4426=2 then do; savingsY=1750; flag4400=1; end;
    if YAS4426=3 then do; savingsY=3750; flag4400=1; end;
    if YAS4426=4 then do; savingsY=7500; flag4400=1; end;
    if YAS4426=5 then do; savingsY=17500; flag4400=1; end;
    if YAS4426=6 then do; savingsY=37500; flag4400=1; end;
    if YAS4426=7 then do; savingsY=50001; flag4400=1; end;
  end;
end;
if YAS4400=-1 or YAS4424L=-1 or YAS4424U=-1 or YAS4426=-1 then savingsY=-1;
if YAS4400=-2 or YAS4424L=-2 or YAS4424U=-2 or YAS4426=-2 then savingsY=-2;
if YAS4400=-3 or YAS4422=-3 or YAS4424L=-3 or YAS4424U=-3 or YAS4426=-3 then savingsY=-3;

/** If youth has other savings in bonds or CDs. */
othsavY=0;
if YAS4530=1 or YAS4530=2 or YAS4530=3 then do;
  if YAS4550=1 and YAS4552>=0 then othsavY=YAS4552;
  if YAS4550=2 and YAS4554U>=0 and YAS4554L>=0 then othsavY=(YAS4554L+(YAS4554U-
YAS4554L)/2);
  if ((YAS4550 ~= 1 and YAS4550 ~= 2) or (YAS4552=-1 or YAS4552=-2)) then do;

```

```

if YAS4556=1 then do; othsavY=500; end;
if YAS4556=2 then do; othsavY=1750; end;
if YAS4556=3 then do; othsavY=3750; end;
if YAS4556=4 then do; othsavY=7500; end;
if YAS4556=5 then do; othsavY=17500; end;
if YAS4556=6 then do; othsavY=37500; end;
if YAS4556=7 then do; othsavY=50001; end;
end;
end;
if YAS4530=-1 or YAS4554L=-1 or YAS4554U=-1 or YAS4556=-1 then othsavY=-1;
if YAS4530=-2 or YAS4554L=-2 or YAS4554U=-2 or YAS4556=-2 then othsavY=-2;
if YAS4530=-3 or YAS4552=-3 or YAS4554L=-3 or YAS4554U=-3 or YAS4556=-3 then othsavY=-3;

/** If youth has stocks, mutual funds. **/
stockY=0;
if YAS4660=1 or YAS4660=2 or YAS4660=3 then do;
  if YAS4680=1 then stockY=YAS4682;
  if YAS4680=2 then stockY=(YAS4684L+(YAS4684U-YAS4684L)/2);
  if ((YAS4680 ~= 1 and YAS4680 ~= 2) or (YAS4682=-1 or YAS4682=-2)) then do;
    if YAS4686=1 then do; stockY=500; end;
    if YAS4686=2 then do; stockY=1750; end;
    if YAS4686=3 then do; stockY=3750; end;
    if YAS4686=4 then do; stockY=7500; end;
    if YAS4686=5 then do; stockY=17500; end;
    if YAS4686=6 then do; stockY=37500; end;
    if YAS4686=7 then do; stockY=50001; end;
  end;
end;
if YAS4660=-1  or YAS4684L=-1 or YAS4684U=-1 or YAS4686=-1 then stockY=-1;
if YAS4660=-2  or YAS4684L=-2 or YAS4684U=-2 or YAS4686=-2 then stockY=-2;
if YAS4660=-3 or YAS4682=-3 or YAS4684L=-3 or YAS4684U=-3 or YAS4686=-3 then stockY=-3;

/** If youth owns vehicles, their PRESENT VALUE. **/
pvcarsY=0;
flag4790=0;
if YAS4790=1 or YAS4790=2 or YAS4790=3 then do;
  if YAS4810=1 then pvcarsY=YAS4812;
  if YAS4810=2 then pvcarsY=(YAS4814L+(YAS4814U-YAS4814L)/2);
  if ((YAS4810 ~= 1 and YAS4810 ~= 2) or (YAS4812=-1 or YAS4812=-2)) then do;
    if YAS4816=1 then do; pvcarsY=2500; flag4790=1; end;
    if YAS4816=2 then do; pvcarsY=7500; flag4790=1; end;
    if YAS4816=3 then do; pvcarsY=17500; flag4790=1; end;
    if YAS4816=4 then do; pvcarsY=37500; flag4790=1; end;
    if YAS4816=5 then do; pvcarsY=75000; flag4790=1; end;
    if YAS4816=6 then do; pvcarsY=175000; flag4790=1; end;
    if YAS4816=7 then do; pvcarsY=250001; flag4790=1; end;
  end;
end;
if YAS4790=-1  or YAS4814L=-1 or YAS4814U=-1 or YAS4816=-1 then pvcarsY=-1;
if YAS4790=-2  or YAS4814L=-2 or YAS4814U=-2 or YAS4816=-2 then pvcarsY=-2;
if YAS4790=-3 or YAS4812=-3 or YAS4814L=-3 or YAS4814U=-3 or YAS4816=-3 then pvcarsY=-3;

```

```

/** If youth still owes money on these vehicles.      */
cardebt=0;
flag4840=0;
if YAS4840=1 then cardebt=YAS4842;
if YAS4840=2 then cardebt=(YAS4844L+(YAS4844U-YAS4844L)/2);
if ((YAS4840 ~= 1 and YAS4840 ~= 2) or (YAS4842=-1 or YAS4842=-2)) then do;
  if YAS4846=1 then do; cardebt=2500; flag4840=1; end;
  if YAS4846=2 then do; cardebt=7500; flag4840=1; end;
  if YAS4846=3 then do; cardebt=17500; flag4840=1; end;
  if YAS4846=4 then do; cardebt=37500; flag4840=1; end;
  if YAS4846=5 then do; cardebt=75000; flag4840=1; end;
  if YAS4846=6 then do; cardebt=175000; flag4840=1; end;
  if YAS4846=7 then do; cardebt=250001; flag4840=1; end;
end;
if YAS4790=-1 or YAS4844L=-1 or YAS4844U=-1 or YAS4846=-1 then cardebt=-1;
if YAS4790=-2 or YAS4844L=-2 or YAS4844U=-2 or YAS4846=-2 then cardebt=-2;
if YAS4790=-3 or YAS4842=-3 or YAS4844L=-3 or YAS4844U=-3 or YAS4846=-3 then cardebt=-3;

/** If youth owns furniture, its PRESENT VALUE */
pvfurntY=0;
if YAS4870=1 then do; pvfurntY=2500; end;
if YAS4870=2 then do; pvfurntY=7500; end;
if YAS4870=3 then do; pvfurntY=17500; end;
if YAS4870=4 then do; pvfurntY=37500; end;
if YAS4870=5 then do; pvfurntY=75000; end;
if YAS4870=6 then do; pvfurntY=175000; end;
if YAS4870=7 then do; pvfurntY=250001; end;
if YAS4870=-1 then pvfurntY=-1;
if YAS4870=-2 then pvfurntY=-2;
if YAS4870=-3 then pvfurntY=-3;

/** If youth owns other assets not mentioned before. */
otassetY=0;
if YAS4880=1 or YAS4880=2 or YAS4880=3 then do;
  if YAS4900=1 then otassetY=YAS4902;
  if YAS4900=2 then otassetY=(YAS4904L+(YAS4904U-YAS4904L)/2);
  if ((YAS4900 ~= 1 and YAS4900 ~= 2) or (YAS4902=-1 or YAS4902=-2)) then do;
    if YAS4906=1 then do; otassetY=2500; end;
    if YAS4906=2 then do; otassetY=7500; end;
    if YAS4906=3 then do; otassetY=17500; end;
    if YAS4906=4 then do; otassetY=37500; end;
    if YAS4906=5 then do; otassetY=75000; end;
    if YAS4906=6 then do; otassetY=175000; end;
    if YAS4906=7 then do; otassetY=250001; end;
  end;
end;
if YAS4880=-1 or YAS4904L=-1 or YAS4904U=-1 or YAS4906=-1 then otassetY=-1;
if YAS4880=-2 or YAS4904L=-2 or YAS4904U=-2 or YAS4906=-2 then otassetY=-2;
if YAS4880=-3 or YAS4902=-3 or YAS4904L=-3 or YAS4904U=-3 or YAS4906=-3 then otassetY=-3;

```

```

/* If youth still owes loans to family or relatives */
ARRAY rloan      (20) rloan01-rloan20;
ARRAY YAS5080  (20) YAS508001-YAS508020;
ARRAY YAS5082  (20) YAS508201-YAS508220;
ARRAY YAS5084L (20) YAS5084L01-YAS5084L20;
ARRAY YAS5084U (20) YAS5084U01-YAS5084U20;
ARRAY YAS5086  (20) YAS508601-YAS508620;
ARRAY YAS5170  (20) YAS517001-YAS517020;
ARRAY YAS5172  (20) YAS517201-YAS517220;
ARRAY YAS5174L (20) YAS5174L01-YAS5174L20;
ARRAY YAS5174U (20) YAS5174U01-YAS5174U20;
ARRAY YAS5176  (20) YAS517601-YAS517620;
ARRAY YAS5074  (20) YAS507401-YAS507420;
ARRAY YAS5066  (20) YAS506601-YAS506620;
ARRAY YAS5164  (20) YAS516401-YAS516420;

do I=1 to DIM(rloan);
  rloan(I)=0;
  if YAS5040=1 and YAS5060 ~= 1 and YAS5062 ~= 1 then do;
    if YAS5080(I)=1 then rloan(I)=YAS5082(I);
    if YAS5080(I)=2 then rloan(I)=(YAS5084L(I)+(YAS5084U(I)-YAS5084L(I))/2);
    if ((YAS5080(I) ~= 1 and YAS5080(I) ~= 2) or (YAS5082(I)=-1 or YAS5082(I)=-2)) then do;
      if YAS5086(I)=1 then do; rloan(I)=500; end;
      if YAS5086(I)=2 then do; rloan(I)=1750; end;
      if YAS5086(I)=3 then do; rloan(I)=3750; end;
      if YAS5086(I)=4 then do; rloan(I)=7500; end;
      if YAS5086(I)=5 then do; rloan(I)=17500; end;
      if YAS5086(I)=6 then do; rloan(I)=37500; end;
      if YAS5086(I)=7 then do; rloan(I)=50001; end;
    end;
  end;
  if YAS5040=-1 or YAS5074(I)=-1 or YAS5086(I)=-1 or YAS5084L[I]=-1 or YAS5084U[I]=-1 then
rloan(I)=-1;
  if YAS5040=-2 or YAS5074(I)=-2 or YAS5086(I)=-2 or YAS5084L[I]=-2 or YAS5084U[I]=-2 then
rloan(I)=-2;
  if YAS5040=-3 or YAS5082(I)=-3 or YAS5086(I)=-3 or YAS5084L[I]=-3 or YAS5084U[I]=-3 then
rloan(I)=-3;

  if YAS5130=1 and YAS5150 ~= 1 and YAS5152 ~= 1 then do;
    if YAS5170(I)=1 then rloan(I)=YAS5172(I);
    if YAS5170(I)=2 then rloan(I)=(YAS5174L(I)+(YAS5174U(I)-YAS5174L(I))/2);
    if ((YAS5170(I) ~= 1 and YAS5170(I) ~= 2) or (YAS5172(I)=-1 or YAS5172(I)=-2)) then do;
      if YAS5176(I)=1 then do; rloan(I)=500; end;
      if YAS5176(I)=2 then do; rloan(I)=1750; end;
      if YAS5176(I)=3 then do; rloan(I)=3750; end;
      if YAS5176(I)=4 then do; rloan(I)=7500; end;
      if YAS5176(I)=5 then do; rloan(I)=17500; end;
      if YAS5176(I)=6 then do; rloan(I)=37500; end;
      if YAS5176(I)=7 then do; rloan(I)=50001; end;
    end;
  end;

```

```

if YAS5130=-1 or YAS5164(I)=-1 or YAS5176(I)=-1 or YAS5174L[I]=-1 or YAS5174U[I]=-1 then
rloan(I)=-1;
if YAS5130=-2 or YAS5164(I)=-2 or YAS5176(I)=-2 or YAS5174L[I]=-2 or YAS5174U[I]=-2 then
rloan(I)=-2;
if YAS5130=-3 or YAS5172(I)=-3 or YAS5176(I)=-3 or YAS5174L[I]=-3 or YAS5174U[I]=-3 then
rloan(I)=-3;
end;

```

```

/** If youth has other debts from loans, credit cards, etc.      */
othdebtY=0;
if YAS5210=1 then do;
  if YAS5220=1 then othdebtY=YAS5222;
  if YAS5220=2 then othdebtY=(YAS5224L+(YAS5224U-YAS5224L)/2);
  if ((YAS5220 ~= 1 and YAS5220 ~= 2) or (YAS5222=-1 or YAS5222=-2)) then do;
    if YAS5226=1 then do; othdebtY=500; end;
    if YAS5226=2 then do; othdebtY=1750; end;
    if YAS5226=3 then do; othdebtY=3750; end;
    if YAS5226=4 then do; othdebtY=7500; end;
    if YAS5226=5 then do; othdebtY=17500; end;
    if YAS5226=6 then do; othdebtY=37500; end;
    if YAS5226=7 then do; othdebtY=50001; end;
  end;
end;
if YAS5210=-1 or YAS5224L=-1 or YAS5224U=-1 or YAS5226=-1 then othdebtY=-1;
if YAS5210=-2 or YAS5224L=-2 or YAS5224U=-2 or YAS5226=-2 then othdebtY=-2;
if YAS5210=-3 or YAS5222=-3 or YAS5224L=-3 or YAS5224U=-3 or YAS5226=-3 then othdebtY=-3;

```

```

/** We now calculate the household net worth according to the youth: hhworthY=assets-liabilities.
*/
/** YAST-105 is the question determining whether respondents are asked the assets question. */

```

```

if (YAS105=0 or YAS105=1) then do;
  hhworthY=0;
  if pvranch NOT in (-1, -2, -3) then hhworthY=hhworthY+pvranch;
  if pvmbst NOT in (-1, -2, -3) then hhworthY=hhworthY+pvmbst;
  if pvmb NOT in (-1, -2, -3) then hhworthY=hhworthY+pvmb;
  if pvst NOT in (-1, -2, -3) then hhworthY=hhworthY+pvst;
  if pvapthm NOT in (-1, -2, -3) then hhworthY=hhworthY+pvapthm;
  if pvbuss NOT in (-1, -2, -3) then hhworthY=hhworthY+pvbuss;
  if secrestY NOT in (-1, -2, -3) then hhworthY=hhworthY+secrestY;
  if retireY NOT in (-1, -2, -3) then hhworthY=hhworthY+retireY;
  if savingsY NOT in (-1, -2, -3) then hhworthY=hhworthY+savingsY;
  if othsavY NOT in (-1, -2, -3) then hhworthY=hhworthY+othsavY;
  if stockY NOT in (-1, -2, -3) then hhworthY=hhworthY+stockY;
  if pvcarsY NOT in (-1, -2, -3) then hhworthY=hhworthY+pvcarsY;
  if pvfurntY NOT in (-1, -2, -3) then hhworthY=hhworthY+pvfurntY;
  if otassetY NOT in (-1, -2, -3) then hhworthY=hhworthY+otassetY;
  if mortgagY NOT in (-1, -2, -3) then hhworthY=hhworthY-mortgagY;
  if loanowed NOT in (-1, -2, -3) then hhworthY=hhworthY-loanowed;
  if stilowed NOT in (-1, -2, -3) then hhworthY=hhworthY-stilowed;

```

```
if secmortY NOT in (-1, -2, -3) then hhworthY=hhworthY-secmortY;
if cardebt NOT in (-1, -2, -3) then hhworthY=hhworthY-cardebt;
if othdebtY NOT in (-1, -2, -3) then hhworthY=hhworthY-othdebtY;
do I=1 to DIM(rloan); if rloan[I] NOT in (-1, -2, -3) then hhworthY=hhworthY-rloan[I]; end;
end;
```

```
if YAS105=-4 then hhworthY=-4;
```

```
IF YAS105=-1 OR (pvranach=-1 OR pvmbst=-1 OR pvmb=-1 OR pvst=-1 OR pvapthm=-1
OR pvbuss=-1 OR secrestY=-1 OR retireY=-1 OR savingsY=-1 OR othsavY=-1 OR stockY=-1
OR pvcarsY=-1 OR pfurntY=-1 OR otassetY=-1 OR mortgagY=-1 OR loanowed=-1
OR stilowed=-1 OR secmortY=-1 OR cardebt=-1 OR rloan01=-1 OR rloan02=-1
OR rloan03=-1 OR rloan04=-1 OR rloan05=-1 OR rloan06=-1 OR rloan07=-1 OR rloan08=-1
OR rloan09=-1 OR rloan10=-1 OR rloan11=-1 OR rloan12=-1 OR rloan13=-1 OR rloan14=-1
OR rloan15=-1 OR rloan16=-1 OR rloan17=-1 OR rloan18=-1 OR rloan19=-1 OR rloan20=-1
OR othdebtY=-1) THEN hhworthY=-1;
```

```
IF YAS105=-2 OR (pvranach=-2 OR pvmbst=-2 OR pvmb=-2 OR pvst=-2 OR pvapthm=-2
OR pvbuss=-2 OR secrestY=-2 OR retireY=-2 OR savingsY=-2 OR othsavY=-2 OR stockY=-2
OR pvcarsY=-2 OR pfurntY=-2 OR otassetY=-2 OR mortgagY=-2 OR loanowed=-2
OR stilowed=-2 OR secmortY=-2 OR cardebt=-2 OR rloan01=-2 OR rloan02=-2
OR rloan03=-2 OR rloan04=-2 OR rloan05=-2 OR rloan06=-2 OR rloan07=-2 OR rloan08=-2
OR rloan09=-2 OR rloan10=-2 OR rloan11=-2 OR rloan12=-2 OR rloan13=-2 OR rloan14=-2
OR rloan15=-2 OR rloan16=-2 OR rloan17=-2 OR rloan18=-2 OR rloan19=-2 OR rloan20=-2
OR othdebtY=-2) THEN hhworthY=-2;
```

```
IF YAS105=-3 OR (pvranach=-3 OR pvmbst=-3 OR pvmb=-3 OR pvst=-3 OR pvapthm=-3
OR pvbuss=-3 OR secrestY=-3 OR retireY=-3 OR savingsY=-3 OR othsavY=-3 OR stockY=-3
OR pvcarsY=-3 OR pfurntY=-3 OR otassetY=-3 OR mortgagY=-3 OR loanowed=-3 OR stilowed=-3
OR secmortY=-3 OR cardebt=-3 OR rloan01=-3 OR rloan02=-3 OR rloan03=-3 OR rloan04=-3
OR rloan05=-3 OR rloan06=-3 OR rloan07=-3 OR rloan08=-3 OR rloan09=-3 OR rloan10=-3
OR rloan11=-3 OR rloan12=-3 OR rloan13=-3 OR rloan14=-3 OR rloan15=-3 OR rloan16=-3
OR rloan17=-3 OR rloan18=-3 OR rloan19=-3 OR rloan20=-3 OR othdebtY=-3) THEN hhworthY=-3;
```

```
/** HANDEDIT: These respondents give a positive value for the lower limit but ZERO for the upper
limit for certain assets. So we give them a -3.    **/
```

```
IF FLAG_RANGE=1 THEN hhworthY=-3;
```

```
/** give the respondents not interviewed in round 6 value -5. **/
```

```
if YAS105=-5 then hhworthY=-5;
```

```
hhworthY=ROUND(hhworthY,1);
```

***** SECTION 3: HOUSEHOLD POVERTY STATUS *****

```
/** First create the poverty thresholds for the youth's household (povert) depending on the household
size and the number of household members under the age of 18 - these values are taken from U.S.
Census' website - http://www.census.gov/hhes/poverty/threshld/thresh01.html; **/
```

```
povert=-4;
```

```
if HHSIZE=-1 or under18=-1 then povert=-1;  
if HHSIZE=-2 or under18=-2 then povert=-2;  
if HHSIZE=-3 or under18=-3 then povert=-3;
```

```
if HHSIZE=1 then povert=9214;
```

```
if HHSIZE=2 then do;
```

```
  if under18=0 then povert=11859;  
  if under18=1 then povert=12207;  
  if under18=2 then povert=12207;
```

```
end;
```

```
if HHSIZE=3 then do;
```

```
  if under18=0 then povert=13853;  
  if under18=1 then povert=14255;  
  if under18=2 then povert=14269;  
  if under18=3 then povert=14269;
```

```
end;
```

```
if HHSIZE=4 then do;
```

```
  if under18=0 then povert=18267;  
  if under18=1 then povert=18566;  
  if under18=2 then povert=17960;  
  if under18=3 then povert=18022;  
  if under18=4 then povert=18022;
```

```
end;
```

```
if HHSIZE=5 then do;
```

```
  if under18=0 then povert=22029;  
  if under18=1 then povert=22349;  
  if under18=2 then povert=21665;  
  if under18=3 then povert=21135;  
  if under18=4 then povert=20812;  
  if under18=5 then povert=20812;
```

```
end;
```

```
if HHSIZE=6 then do;
```

```
  if under18=0 then povert=25337;  
  if under18=1 then povert=25438;  
  if under18=2 then povert=24914;  
  if under18=3 then povert=24411;  
  if under18=4 then povert=23664;  
  if under18=5 then povert=23221;  
  if under18=6 then povert=23221;
```

```
end;
```

```
if HHSIZE=7 then do;
```

```
  if under18=0 then povert=29154;  
  if under18=1 then povert=29336;  
  if under18=2 then povert=28708;  
  if under18=3 then povert=28271;
```

```
if under18=4 then povert=27456;
if under18=5 then povert=26505;
if under18=6 then povert=25462;
if under18=7 then povert=25462;
end;
```

```
if HHSIZE=8 then do;
  if under18=0 then povert=32606;
  if under18=1 then povert=32894;
  if under18=2 then povert=32302;
  if under18=3 then povert=31783;
  if under18=4 then povert=31047;
  if under18=5 then povert=30112;
  if under18=6 then povert=29140;
  if under18=7 then povert=28893;
  if under18=8 then povert=28893;
end;
```

```
if HHSIZE>=9 then do;
  if under18= 0 then povert=39223;
  if under18= 1 then povert=39413;
  if under18= 2 then povert=38889;
  if under18= 3 then povert=38449;
  if under18= 4 then povert=37726;
  if under18= 5 then povert=36732;
  if under18= 6 then povert=35833;
  if under18= 7 then povert=35610;
  if under18>=8 then povert=34238;
end;
```

*** Then create the poverty ratio (povthr) which is the gross household income of the youth (groshhIY) divided by the poverty threshold relevant to the youth's household (povert);

```
povthr=-4;
if groshhIY>=0 and povert>0 then povthr=ROUND(((groshhIY*100)/povert),1);
if groshhIY eq -1 or povert=-1 then povthr=-1;
if groshhIY eq -2 or povert=-2 then povthr=-2;
if groshhIY eq -3 or povert=-3 then povthr=-3;
if groshhIY eq -4 then povthr=-4;
```

```
/** give respondents who were not interviewed in round 6 a value of -5    */
if YAS050=-5 then povthr=-5;
```

```
/** Changing all invalid skips to -3.      */
if groshhIY=-1 then groshhIY=-3; if groshhIY=-2 then groshhIY=-3;
if hhworthY=-1 then hhworthY=-3; if hhworthY=-2 then hhworthY=-3;
if povthr=-1 then povthr=-3; if povthr=-2 then povthr=-3;
```

```
IF groshhIY>=185000 THEN DO;
  groshhIY=267926.78;
```

```
topGIY=1;
END;
IF hhworthY>=75700 THEN DO;
hhworthY=201760.77;
topHHY=1;
END;
IF povthr>=1094 THEN DO;
povthr=1520.86;
topPOV=1;
END;
```

***** SECTION 4: TOPCODING INCOME, NET WORTH, AND POVERTY RATIO *****

```
/** Topcoding the groshhIY, hhworthY and povthr variables.      **/
```

```
topcode_GIY=191001;
topcode_HHY=183900;
topcode_POV=1151;
```

```
mean_topcode_GIY=285805;
mean_topcode_HHY=1111342;
mean_topcode_POV=1720;
```

```
if groshhIY>=topcode_GIY then do; groshhIY=mean_topcode_GIY; topcode1=1; end;
if hhworthY>=topcode_HHY then do; hhworthY=mean_topcode_HHY; topcode2=1; end;
if povthr>=topcode_POV then do; povthr=mean_topcode_POV; topcode3=1; end;
```

```
if topcode1=1 and topcode2=. and topcode3=. then topcode=1;
if topcode1=. and topcode2=1 and topcode3=. then topcode=2;
if topcode1=. and topcode2=. and topcode3=1 then topcode=3;
```

```
if topcode1=1 and topcode2=1 and topcode3=. then topcode=12;
if topcode1=1 and topcode2=. and topcode3=1 then topcode=13;
if topcode1=. and topcode2=1 and topcode3=1 then topcode=23;
```

```
if topcode1=1 and topcode2=1 and topcode3=1 then topcode=123;
```

```
endsas;
```

[Return to top](#)

Participation in Government Programs

Variables Created:

- CV_AMT_GOVNT_PCM_PCY.80 - CV_AMT_GOVNT_PCM_PCY.99
CV_AMT_GOVNT_PCM_PCY.00 - CV_AMT_GOVNT_PCM_PCY.03
- CV_GOVNT_PRG_EVER
- CV_GOVNT_PRG_YR.80 - CV_GOVNT_PRG_YR.99 CV_GOVNT_PRG_YR.00 -
CV_GOVNT_PRG_YR.03

Variables Used

Please contact NLS User Services for variable names.

This program creates several variables describing the respondent's participation in government programs for the economically disadvantaged. During the interview, respondents report amounts received and months of participation in Aid to Families with Dependent Children (AFDC); food stamps; and Women, Infants, and Children (WIC). There is also an "other assistance" question to capture information about any other government program from which respondents may have received assistance. The interview also records amounts received from unemployment compensation and worker's compensation in separate question series; this program also creates variables describing respondents' participation in unemployment or worker's compensation.

The program to create these variables first creates a month-by-month participation array for five of the six categories (AFDC, food stamps, WIC, other programs, and unemployment compensation; the array is not created for worker's compensation). These month-by-month variables constitute part of the event history array for program participation; see [appendix 7](#) for more information. After all five arrays are created, the program merges data to create the summary variables.

Because worker's compensation can be reported either as a lump sum received on a single day or as payments over the period of time that the respondent was not working, the month-by-month array variables are no longer created for this category of program participation. The total benefits received in each calendar year are calculated, but the weeks and months that compensation were received are not created.

***** SECTION 1: UNEMPLOYMENT COMPENSATION *****

Variables created:

```

CV_UI_EVER
CV_UI_YR.80-CV_UI_YR.99 CV_UI_YR.00 CV_UI_YR.01 CV_UI_YR.02
CV_UI_YR.03
CV_AMT_UI_YR.80-CV_AMT_UI_YR.99 CV_AMT_UI_YR.00
CV_AMT_UI_YR.01
CV_AMT_UI_YR.02 CV_AMT_UI_YR.03
CV_UI_SPELLS_YR.80-CV_UI_SPELLS_YR.99 CV_UI_SPELLS_YR.00
CV_UI_SPELLS_YR.01
CV_UI_SPELLS_YR.02 CV_UI_SPELLS_YR.03

```

/** Almost all arrays used in this program are as follows:

**/

```

array intd      (5)  int01_d int00_d int99_d int98_d int97_d;
array intm      (5)  int01_m int00_m int99_m int98_m int97_m;

```

```

array inty      (5) int01_d int00_y int99_y int98_y int97_y;
array ui_everpr (5) ui5_ever ui4_ever ui3_ever ui2_ever ui1_ever;

array uitotalpr (5,24) ui5total80-ui5total99   ui5total00  ui5total01  ui5total02  ui5total03
                  ui4total80-ui4total99   ui4total00  ui4total01  ui4total02  ui4total03
                  ui3total80-ui3total99   ui3total00  ui3total01  ui3total02  ui3total03
                  ui2total80-ui2total99   ui2total00  ui2total01  ui2total02  ui2total03
                  ui1total80-ui1total99   ui1total00  ui1total01  ui1total02  ui1total03;

array uiatotalpr (5,24) uia5total80-uaia5total99  uia5total00 uia5total01 uia5total02 uia5total03
                  uia4total80-uaia4total99  uia4total00 uia4total01 uia4total02 uia4total03
                  uia3total80-uaia3total99  uia3total00 uia3total01 uia3total02 uia3total03
                  uia2total80-uaia2total99  uia2total00 uia2total01 uia2total02 uia2total03
                  uaia1total80-uaia1total99 uaia1total00 uaia1total01 uaia1total02 uaia1total03;

array ui_spellspr (5,24) ui5_spells80-ui5_spells99  ui5_spells00 ui5_spells01 ui5_spells02
ui5_spells03
                  ui4_spells80-ui4_spells99  ui4_spells00 ui4_spells01 ui4_spells02 ui4_spells03
                  ui3_spells80-ui3_spells99  ui3_spells00 ui3_spells01 ui3_spells02 ui3_spells03
                  ui2_spells80-ui2_spells99  ui2_spells00 ui2_spells01 ui2_spells02 ui2_spells03
                  ui1_spells80-ui1_spells99  ui1_spells00 ui1_spells01 ui1_spells02 ui1_spells03;

array uitotal    (24) uitotal80-uitotal99      uitotal00  uitotal01  uitotal02  uitotal03;
array uiatotal   (24) uiatotal80-uaiatotal99  uiatotal00  uiatotal01  uiatotal02  uiatotal03;
array ui_spells  (24) ui_spells80-ui_spells99 ui_spells00 ui_spells01 ui_spells02 ui_spells03;
array nu        (24) nu80-nu99  nu00  nu01  nu02  nu03;

array ouitotal   (24) ouitotal80-ouitotal99  ouitotal00  ouitotal01  ouitotal02  ouitotal03;
array ouiatotal  (24) ouiatotal80-ouiatotal99 ouiatotal00  ouiatotal01  ouiatotal02  ouiatotal03;
array oui_spells (24) oui_spells80-oui_spells99 oui_spells00 oui_spells01 oui_spells02 oui_spells03;
array CURRnu    (24) CURRnu80-CURRnu99 CURRnu00 CURRnu01 CURRnu02 CURRnu03;

array um        (24) um80-um99   um00  um01  um02  um03;
array uamt      (24) uamt80-uamt99 uamt00 uamt01 uamt02 uamt03;
array ua        (24) ua80-ua99   ua00  ua01  ua02  ua03;
array un        (24) un80-un99   un00  un01  un02  un03;

array ui        (288) ui001-ui288;
array uiamt     (288) uiamt001-uiamt288;

array yms       (2,4) ymsu11-ymsu14 ymsu21-ymsu24;
array csm       (2,4) csmu11-csmu14 csmu21-csmu24;
array yme       (2,4) ymeu11-ymeu14 ymeu21-ymeu24;
array cem       (2,4) cemu11-cemu14 cemu21-cemu24;
array s_day     (2,4) sdayu11-sdayu14 sdayu21-sdayu24;
array e_day     (2,4) edayu11-edayu14 edayu21-edayu24;
array mos       (2,4) mosu11-mosu14 mosu21-mosu24;
array nms       (2,4) nmsu11-nmsu14 nmsu21-nmsu24;
array nme       (2,4) nmeu11-nmeu14 nmeu21-nmeu24;
array newms    (2,4) newms11-newms14 newms21-newms24;
array newme    (2,4) newme11-newme14 newme21-newme24;

```

```

array eflag      (2,4) eflagu11-eflagu14 eflagu21-eflagu24;
array Neflag     (2,4) Neflagu11-Neflagu14 Neflagu21-Neflagu24;
array efl        (2,4) eflu11-eflu14 eflu21-eflu24;
array cembfdli  (2,4) cembfdliu11-cembfdliu14 cembfdliu21-cembfdliu24;
array incc       (2,4) inccu11-inccu14 inccu21-inccu24;
array aflag      (2,4) aflagu11-aflagu14 aflagu21-aflagu24;
array agefl      (2,4) ageflu11-ageflu14 ageflu21-ageflu24;

/** start YEAR, MONTH, ACTUAL DAY and ESTIMATED DAY
information.                                     */
array ys        (2,4) ps40001Y ps40002Y ps40003Y ps40004Y p40001Y p40002Y p40003Y
p40004Y;
array ms        (2,4) ps40001M ps40002M ps40003M ps40004M p40001M p40002M p40003M
p40004M;
array ds        (2,4) ps42001   ps42002   ps42003   ps42004   p42001   p42002   p42003
p42004;
array es        (2,4) ps43001   ps43002   ps43003   ps43004   p43001   p43002   p43003
p43004;

/** end YEAR, MONTH, ACTUAL DAY and ESTIMATED DAY
information.                                     */
array ye        (2,4) ps47001Y ps47002Y ps47003Y ps47004Y p47001Y p47002Y p47003Y
p47004Y;
array me        (2,4) ps47001M ps47002M ps47003M ps47004M p47001M p47002M p47003M
p47004M;
array de        (2,4) ps49001   ps49002   ps49003   ps42004   p49001   p49002   p49003
p49004;
array ee        (2,4) ps50001   ps50002   ps50003   ps50004   p50001   p50002   p50003
p50004;

/** ESTIMATED WEEKS - unable to answer start or stop date questions.           */
array wks       (2,4) ps52001   ps52002   ps52003   ps52004   p52001   p52002   p52003   p52004;

/** CURRENTLY receiving information:
For NEVER PRIOR sections: Since respondents are routed through different branches some are asked
the
currently receiving information in PS1897 and others in p35001. The information from ps1897 and
p35001 are collected into p35001 later in the program.;*/
array cur       (2,4) ps44001   ps44002   ps44003   ps44004   p44001   p44002   p44003   p44004;
array cwks      (2,4) ps19301   ps19302   ps19303   ps19304   p35001   p35002   p35003   p35004;

/** Receiving at DLI - using ps18501 instead of ps18301 as the latter is a computer check from last
interview and the former is the follow up question that verifies that information.          */
array dli       (2,4) ps18501   ps18502   ps18503   ps18504   p18501   p18502   p18503   p18504;

/** AMOUNT RECEIVED - actual & estimated.                                         */
array inc       (2,4) ps59001   ps59002   ps59003   ps59004   p59001   p59002   p59003   p59004;
array einc      (2,4) ps60001   ps60002   ps60003   ps60004   p60001   p60002   p60003   p60004;

```

```
** Part 1. Creating certain basic variables
```

```
*****;
```

```
/** change weekly ui figures to daily amount.
```

```
**/
```

```
do k=1 to dim(inc,1);  
do J=1 to dim(inc,2);  
if inc(k,j)>0 then incc(k,j)=inc(k,j)/7;  
if inc(k,j)<=0 then incc(k,j)=inc(k,j);  
end;  
end;
```

```
/** change previous round variables to account for noninterviews -
```

```
If interviewed in Round 5 then use that. If non-interview in Round5, then use Round4 variable....  
and so on.;
```

```
**/
```

```
do i=1 to dim(UITOTAL);  
ouitotal(i)=UITOTAL(i);  
ouiatotal(i)=UIATOTAL(i);  
oui_spells(i)=ui_spells(i);  
NU(i)=0;
```

```
if UITOTAL(i)=-. then do;
```

```
ouitotal(i)=-4;
```

```
UITOTAL(i)=0;
```

```
end;
```

```
if UIATOTAL(i)=-. then do;
```

```
ouiatotal(i)=-4;
```

```
UIATOTAL(i)=0;
```

```
end;
```

```
if ui_spells(i)=-. then do;
```

```
oui_spells(i)=-4;
```

```
ui_spells(i)=0;
```

```
end;
```

```
end;
```

```
if ui_ever=-4 then do;
```

```
ui_ever=0;
```

```
end;
```

```
** Initialize the following counters/dummy variables to zero. ;
```

```
** uick: Counter for UIAMT>=0 (receiving amount info. valid) but UI=0 (not receiving).;
```

```
** uickm: Counter for UI>0 (receiving) but UIAMT is missing (amount receiving info. not available).
```

```
** uickmm: Counter for UI=-3 (receiving info. invalid) but UIAMT~-=-3 (amount receiving info valid).
```

```
** TTLM: Total number of months ever received.;
```

```
** uimnths: Month counter for correction of problematic (too high) amounts reported.;
```

```
** SAMEMONTH: Dummy variable to indicate that start and stop months are the same.;
```

```
uick=0;  
uickm=0;  
uickmm=0;  
ttl=0;  
uimnths=0;
```

SAMEMONTH=0;

** Create variables relating to date of birth, age, age 14, current interview, DLI etc.:

ymb - Date of Birth in YYYYMM scheme

cmb - Date of Birth in continuous month scheme (CMS)

ym14 - Date of 14th birthday in YYYYMM scheme

cm14 - Date of 14th birthday in CMS

iym - Date of Current Interview(DCI) in YYYYMM scheme

doicm - DCI in CMS

dliy - DLI in YYYYMM scheme

dlicm - DLI in CMS

dli_d - day of DLI

dli_m - month of DLI

dli_y - year of DLI

uiday - show the number of days in a particular month. February is taken to have 28 days irrespective of whether it is a leap year.

ddliday - number of days in the DLI month after the DLI day including the DLI day. When we are imputing start dates and impute a start date to be DLI, then we

use the number of days in that month from onwards DLI day as the days receiving by the respondent.

dliday - number of days in the DLI month after the DLI day (dliday1) plus the number of days in the following month (dliday2). If a respondent reports receiving at

the date of current interview then we count the number of days receiving till the date of current interview. In the next interview if the respondent reports

receiving since DLI then take the month following the DLI month as the start month for the spell for the next round. This artificial spell start date results in

not counting the days since DLI to a month after that. For example if DLI is May 1, 2002, we would count the next spell starting from June 1, 2002 and

continuing till July 31, 2002. We would erroneously code the respondent as not having received anything for 1 month (between May 1, 2002 and June 1,

2002). To correct for this, if a respondent reports receiving SDLI we use DLIDAY as the number of days he/she received in the first month of this new

spell. Continuing with our example, this respondent will show as receiving for 70 days for June of 2002, dliday1=30, dliday2=30 and dliday=70.;

array mon (12) mon1-mon12;

array uiday (12) uiday1-uiday12;

ymb=(bdate_y*100+bdate_m);

cmb=((bdate_y)-1980)*12+bdate_m;

ym14=(bdate_y*100+bdate_m)+1400;

cm14=((bdate_y+14)-1980)*12+bdate_m;

if yint_y>0 and yint_m>0 then iym=yint_y*100+yint_m;

if yint_y in (-1,-2,-3) or yint_m in (-1,-2,-3) then iym=-3;

if yint_y>0 and yint_m>0 then doicm=(yint_y-1980)*12+yint_m;

if yint_y in (-1,-2,-3) or yint_m in (-1,-2,-3) then doicm=-3;

```

do Z=1 to 12; mon(Z)=Z; end;

do Z=1 to 12;
  if mon(Z)=1 OR mon(Z)=3 OR mon(Z)=5 OR mon(Z)=7 OR mon(Z)=8 OR mon(Z)=10 OR mon(Z)=12 then do;
    uiday(Z)=31;
  end;
  if mon(Z)=2 then do;
    uiday(Z)=28;
  end;
  if mon(Z)=4 OR mon(Z)=6 OR mon(Z)=9 OR mon(Z)=11 then do;
    uiday(Z)=30;
  end;
end;

/** Creating variables pertaining to the date of last interview. */
if p1210=1 then do;
  do q=1 to dim(inty);
    if inty[q]~=-5 then do;
      if intm[q]<12 then dliym=(inty[q]*100)+(intm[q]+1);
      if intm[q]=12 then dliym=((inty[q]+1)*100)+1;
      dlicm=((inty[q]-1980)*12)+(intm[q]+1);
      dli_d=intd[q];
      dli_m=dliym-round(dliym,100);
      dli_y=round(dliym,100)/100;
      q=dim(inty);
    end;
  end;
end;

if p1210~=1 then do;
  dliym=ym14;
  dlicm=cm14;
  dli_d=bdate_d;
  dli_m=bdate_m;
  dli_y=bdate_y;
end;

do z=1 to 12;
if dli_m=mon(z) then ddiday=uiday(z)+1-dli_d;
end;

if dli_m=1 or dli_m=8 then do;
  ddiday=(31-dli_d+31);
  ddiday1=31-dli_d;
  ddiday2=31;
end;
if dli_m=3 then do;
  ddiday=(28-dli_d+31);

```

```
dliday1=28-dli_d;  
dliday2=31;
```

```
end;
```

```
if dli_m in (5, 7, 10, 12) then do;
```

```
    dliday=(30-dli_d+31);
```

```
    dliday1=30-dli_d;
```

```
    dliday2=31;
```

```
end;
```

```
if dli_m in (4, 6, 9, 11) then do;
```

```
    dliday=(31-dli_d+30);
```

```
    dliday1=31-dli_d;
```

```
    dliday2=30;
```

```
end;
```

```
if dli_m=2 then do;
```

```
    dliday=(31-dli_d+28);
```

```
    dliday1=31-dli_d;
```

```
    dliday2=28;
```

```
end;
```

```
ps19302=ps19301;
```

```
ps19303=ps19301;
```

```
ps19304=ps19301;
```

```
p35002=p35001;
```

```
p35003=p35001;
```

```
p35004=p35001;
```

```
** Part 2. Creating variables using Round 6 information
```

```
*****;
```

```
if p1210>-4 then do; 
```

```
        /** [B1] **/
```

```
/** Combine currently receiving information for the NEVER PRIOR sections. */
```

```
if ps1897>-3 then do;
```

```
    p35001=ps1897;
```

```
    p35002=p35001;
```

```
    p35003=p35001;
```

```
end;
```

```
/** Initialize all ui from the month of birth to the date of current interview to zero. */
```

```
do L=1 to DIM(ui);
```

```
    if cmb<=L<=doicm then do;
```

```
        ui(L)=0;
```

```
    end;
```

```
end;
```

```
/** This portion of the SAS program defines the start date and end dates. If the respondent reports still receiving, the interview date is used as the temporary end date for the last loop reported.
```

```
In the next survey round, the respondent will be asked if he or she is still receiving and, if not, a permanent end date equivalent to the interview date of the previous round will be assigned. Users will be able to tell which method was used by looking at the following participation flag variable created during the program. The categories are the following:
```

1=respondent reported participation dates
 2=start month imputed
 3=start month and year imputed
 4=stop month imputed
 5=stop month and year imputed
 6=start and stop dates imputed
 7=error in data due to round 5 questionnaire problem or inconsistent response. Both results in our
 not sure whether the respondents received in these months
 or not.

Questionnaire Error: These respondents reported valid start and stop dates but were never asked the follow-up question - if they received again.

Conflicting Response: These respondents reported currently receiving, a valid start date and a valid stop date that is different from the current interview

date but in the follow-up question as to whether they received again, they answered NO.

******/

```

do K=1 to DIM(csm,1);                                /** [B2] **/  

  do J=1 to DIM(csm,2);                            /** [B3] **/  

  

/** 1. If start and stop dates are all known          **/  

  if ys(K,J)>0 and ms(K,J)>0 then do;  

    yms(K,J)=(ys(K,J)*100)+ms(K,J);  

  

  /** set start day if known. If unknown then use estimated period of the month when respondent  

  started receiving**/  

  if ds(K,J)>0 OR es(K,J)>0 then do;  

    if ms(K,J)=1 OR ms(K,J)=3 OR ms(K,J)=5 OR ms(K,J)=7 OR ms(K,J)=8 OR ms(K,J)=10 OR  

    ms(K,J)=12  

    then do;  

      if ds(K,J)>0 and ds(K,J) ~=. then s_day(K,J)=(32-ds(K,J));  

      if es(K,J)=1 then s_day(K,J)=31;  

      if es(K,J)=2 then s_day(K,J)=21;  

      if es(K,J)=3 then s_day(K,J)=11;  

    end;  

    if ms(K,J)=4 OR ms(K,J)=6 OR ms(K,J)=9 OR ms(K,J)=11 then do;  

      if ds(K,J)>0 and ds(K,J) ~=. then s_day(K,J)=(31-ds(K,J));  

      if es(K,J)=1 then s_day(K,J)=30;  

      if es(K,J)=2 then s_day(K,J)=20;  

      if es(K,J)=3 then s_day(K,J)=10;  

    end;  

    if ms(K,J)=2 then do;  

      if ds(K,J)>0 and ds(K,J) ~=. then s_day(K,J)=(29-ds(K,J));  

      if es(K,J)=1 then s_day(K,J)=28;  

      if es(K,J)=2 then s_day(K,J)=18;  

      if es(K,J)=3 then s_day(K,J)=8;  

    end;  

  end;  

end;  

  

/** set start date to DLI if receiving at DLI.          **/  

if dli(K,J)=1 then do;

```

```

yms(K,J)=dliym;
s_day(K,J)=dliday;
if ps19301=1 then do;
  yme(K,J)=iym;
  e_day(K,J)=yint_d;
  NOGAPS=1;
end;
end;

** set end date to DCI if receiving at DCI.;

if cur(k,j)=1 then do;
  yme(k,j)=iym;
  e_day(k,j)=yint_d;
end;
end;

/** set end date if both end month and year are known. */
if ye(K,J)>0 and me(K,J)>0 then do;
  yme(K,J)=(ye(K,J)*100)+me(K,J);

/** set end day, if known. */
if de(K,J)>0 and de(K,J) ~= . then do;
  e_day(K,J)=de(K,J);
end;

/** if end day is unknown, then use estimate of the period of the month when respondent stopped
receiving; */
if de(K,J)<0 and ee(K,J)>0 and ee(K,J) ~= . then do;
  if ee(K,J)=1 then e_day(K,J)=9;
  if ee(K,J)=2 then e_day(K,J)=19;
  if ee(K,J)=3 then e_day(K,J)=28;
end;
end;

/** if end month greater than interview month, constrain by interview month.
*/
if yme(K,J)>iym then do;
  yme(K,J)=iym;
  e_day(K,J)=yint_d;
end;
if yms(K,J)>0 and yme(K,J)>=dliym then eflag(K,J)=1;
if yms(k,j)>0 and yme(k,j)<dliym then Neflag(k,j)=1;

/** determine months from estimated weeks variable. */
if wks(K,J)>=0 and (ys(K,J)<0 OR ms(K,J)<0 OR ye(K,J)<0 OR me(K,J)<0) then do;
  mos(K,J)=floor(wks(K,J)/4.3);
end;
agefl(K,J)=0;
if ys(K,J)>0 and ys(K,J)<1990 then agefl(K,J)=1;

```

```

/** 2. If start year is known and month is unknown */
if ys(K,J)>0 and -3 <= ms(K,J) <= -1 then do;

/** 2.a If estimated weeks are known and currently receiving, then count backwards by the number of
weeks from DCI. If the number of weeks falls short of the
start year, then set start month to December of the start year. If the number of estimated weeks is
past the start year, then set start month to January of the
start year. If start date is before DLI, then constrain by DLI.; **/


if (wks(K,J) >= 0 and cwks(K,J)=1) then do;
  yme(K,J)=iym;
  e_day(K,J)=yint_d;

  if (      0)<=mos(K,J)<=(yint_m- 1) then yms(K,J)=yme(K,J)- 0-(mos(K,J)- 0);
  if (yint_m+ 0)<=mos(K,J)<=(yint_m+ 11) then yms(K,J)=yme(K,J)- 100-(mos(K,J)- 12);
  if (yint_m+ 12)<=mos(K,J)<=(yint_m+ 23) then yms(K,J)=yme(K,J)- 200-(mos(K,J)- 24);
  if (yint_m+ 24)<=mos(K,J)<=(yint_m+ 35) then yms(K,J)=yme(K,J)- 300-(mos(K,J)- 36);
  if (yint_m+ 36)<=mos(K,J)<=(yint_m+ 47) then yms(K,J)=yme(K,J)- 400-(mos(K,J)- 48);
  if (yint_m+ 48)<=mos(K,J)<=(yint_m+ 59) then yms(K,J)=yme(K,J)- 500-(mos(K,J)- 60);
  if (yint_m+ 60)<=mos(K,J)<=(yint_m+ 71) then yms(K,J)=yme(K,J)- 600-(mos(K,J)- 72);
  if (yint_m+ 72)<=mos(K,J)<=(yint_m+ 83) then yms(K,J)=yme(K,J)- 700-(mos(K,J)- 84);
  if (yint_m+ 84)<=mos(K,J)<=(yint_m+ 95) then yms(K,J)=yme(K,J)- 800-(mos(K,J)- 96);
  if (yint_m+ 96)<=mos(K,J)<=(yint_m+107) then yms(K,J)=yme(K,J)- 900-(mos(K,J)-108);
  if (yint_m+108)<=mos(K,J)<=(yint_m+119) then yms(K,J)=yme(K,J)-1000-(mos(K,J)-120);
  if (yint_m+120)<=mos(K,J)<=(yint_m+131) then yms(K,J)=yme(K,J)-1100-(mos(K,J)-132);

  do Z=1 to 12;
    if (yms(K,J)-(round(yms(K,J), 100)))=mon(Z) then s_day(K,J)=uiday(Z);
  end;

  if yms(K,J)<((ys(K,J)*100)+01) then do;
    yms(K,J)=((ys(K,J)*100)+01);
    s_day(K,J)=31;
  end;

  if yms(K,J)<dliym then do;
    yms(K,J)=dliym;
    s_day(K,J)=ddliday;
  end;

  if yms(K,J)>((ys(K,J)*100)+12) then do;
    yms(K,J)=((ys(K,J)*100)+12);
    s_day(K,J)=31;
  end;
end;

/** 2.b If the estimated weeks are missing and currently receiving then set end date to current interview
date and start date to December of the start year and
constrain start date by DCI and DLI.; **/


else if (-3<=wks(K,J)<0 and cwks(K,J)=1) then do;

```

```

yme(K,J)=iym;
e_day(K,J)=yint_d;
yms(K,J)=((ys(K,J)*100)+12);
s_day(K,J)=31;

if yms(K,J)>iym then do;
  yms(K,J)=iym;
  s_day(K,J)=intday;
end;

if yms(K,J)<dliym then do;
  yms(K,J)=dliym;
  s_day(K,J)=ddliday;
end;

end;

```

/** 2.c If estimated weeks are missing and not currently receiving then set start month to January of the start year and end date to December of the start year.

Constrain start date by DLI and end date by current interview.; **/

else if (-3<=wks(K,J)<0 and cwks(K,J)=0) then do;

```

yms(K,J)=((ys(K,J)*100)+01);
s_day(K,J)=31;
yme(K,J)=((ys(K,J)*100)+12);
e_day(K,J)=31;

```

```

if yms(K,J)<dliym then do;
  yms(K,J)=dliym;
  s_day(K,J)=ddliday;
end;

```

```

if yme(K,J)>iym then do;
  yme(K,J)=iym;
  e_day(K,J)=yint_d;
end;
end;

```

/** 2.d If estimated weeks are known and not currently receiving, then count forward by the number of estimated weeks from January of the start year and set end

date. Constrain start date by DLI. If the count exceeds the interview date then stop counting at the interview date.; **/

else if (wks(K,J)>=0 and cwks(K,J)=0) then do;

```

yms(K,J)=((ys(K,J)*100)+01);
s_day(K,J)=31;
if yms(K,J)<dliym then do;
  yms(K,J)=dliym;

```

```

s_day(K,J)=ddliday;
end;

Nms(k,j)=yms(k,j)- round(yms(k,j),100);

if ( 0)<=mos(k,j)<=(12-Nms(k,j)+ 0) then yme(k,j)=yms(k,j)+mos(k,j);
if (12-Nms(k,j)+ 1)<=mos(k,j)<=(12-Nms(k,j)+ 12) then yme(k,j)=yms(k,j)+ 100+mos
(k,j)- 12;
if (12-Nms(k,j)+ 13)<=mos(k,j)<=(12-Nms(k,j)+ 24) then yme(k,j)=yms(k,j)+ 200+mos
(k,j)- 24;
if (12-Nms(k,j)+ 25)<=mos(k,j)<=(12-Nms(k,j)+ 36) then yme(k,j)=yms(k,j)+ 300+mos
(k,j)- 36;
if (12-Nms(k,j)+ 37)<=mos(k,j)<=(12-Nms(k,j)+ 48) then yme(k,j)=yms(k,j)+ 400+mos
(k,j)- 48;
if (12-Nms(k,j)+ 49)<=mos(k,j)<=(12-Nms(k,j)+ 60) then yme(k,j)=yms(k,j)+ 500+mos
(k,j)- 60;
if (12-Nms(k,j)+ 61)<=mos(k,j)<=(12-Nms(k,j)+ 72) then yme(k,j)=yms(k,j)+ 600+mos
(k,j)- 72;
if (12-Nms(k,j)+ 73)<=mos(k,j)<=(12-Nms(k,j)+ 84) then yme(k,j)=yms(k,j)+ 700+mos
(k,j)- 84;
if (12-Nms(k,j)+ 85)<=mos(k,j)<=(12-Nms(k,j)+ 96) then yme(k,j)=yms(k,j)+ 800+mos
(k,j)- 96;
if (12-Nms(k,j)+ 97)<=mos(k,j)<=(12-Nms(k,j)+108) then yme(k,j)=yms(k,j)+ 900+mos
(k,j)-108;
if (12-Nms(k,j)+109)<=mos(k,j)<=(12-Nms(k,j)+120) then yme(k,j)=yms(k,j)+1000+mos
(k,j)-120;
if (12-Nms(k,j)+121)<=mos(k,j)<=(12-Nms(k,j)+132) then yme(k,j)=yms(k,j)+1100+mos
(k,j)-132;

do z=1 to 12;
if (yme(k,j)-(round(yme(k,j), 100)))=mon(z) then e_day(k,j)=uiday(z);
end;

if yme(k,j)>iym then do;
yme(k,j)=iym;
e_day(k,j)=yint_d;
end;

end;

if yme(k,j)>0 and wks(k,j)=-4 then do;
yms(k,j)=((ys(k,j)*100)+01);
s_day(k,j)=31;
end;

if yms(k,j)<dliym then do;
yms(k,j)=dliym;
s_day(k,j)=ddliday;
end;

if yms(k,j)>((ys(k,j)*100)+12) then do;

```

```

yms(k,j)=((ys(k,j)*100)+12);
s_day(k,j)=31;
end;

eflag(k,j)=2;

end;

/** 3. If start year is unknown. */  

if -3<=ys(K,J)<=-1 then do;  

  

/** 3.a If currently receiving and weeks are known then count back from interview date receiving. */
if (wks(K,J)>=0 and cwks(K,J)=1) then do;  

  

  if (      0)<=mos(K,J)<=(yint_m- 1) then yms(K,J)=iym- 0-(mos(K,J)- 0);  

  if (yint_m+ 0)<=mos(K,J)<=(yint_m+ 11) then yms(K,J)=iym- 100-(mos(K,J)- 12);  

  if (yint_m+ 12)<=mos(K,J)<=(yint_m+ 23) then yms(K,J)=iym- 200-(mos(K,J)- 24);  

  if (yint_m+ 24)<=mos(K,J)<=(yint_m+ 35) then yms(K,J)=iym- 300-(mos(K,J)- 36);  

  if (yint_m+ 36)<=mos(K,J)<=(yint_m+ 47) then yms(K,J)=iym- 400-(mos(K,J)- 48);  

  if (yint_m+ 48)<=mos(K,J)<=(yint_m+ 59) then yms(K,J)=iym- 500-(mos(K,J)- 60);  

  if (yint_m+ 60)<=mos(K,J)<=(yint_m+ 71) then yms(K,J)=iym- 600-(mos(K,J)- 72);  

  if (yint_m+ 72)<=mos(K,J)<=(yint_m+ 83) then yms(K,J)=iym- 700-(mos(K,J)- 84);  

  if (yint_m+ 84)<=mos(K,J)<=(yint_m+ 95) then yms(K,J)=iym- 800-(mos(K,J)- 96);  

  if (yint_m+ 96)<=mos(K,J)<=(yint_m+107) then yms(K,J)=iym- 900-(mos(K,J)-108);  

  if (yint_m+108)<=mos(K,J)<=(yint_m+119) then yms(K,J)=iym-1000-(mos(K,J)-120);  

  if (yint_m+120)<=mos(K,J)<=(yint_m+131) then yms(K,J)=iym-1100-(mos(K,J)-132);  

  

  do Z=1 to 12;  

    if yms(K,J)-(round(yms(K,J),100))=mon(Z) then s_day(K,J)=uiday(Z);  

  end;  

  

  yme(K,J)=iym;  

  e_day(K,J)=yint_d;  

  if yms(K,J)<dliym then do;  

    yms(K,J)=dliym;  

    s_day(K,J)=ddliday;  

  end;  

end;  

  

/** 3.b If currently receiving and estimated weeks are unknown then set start date to DLI and stop date to current interview. */
if (-3<=wks(K,J)<0 and cwks(K,J)=1) then do;  

  yms(K,J)=dliym;  

  yme(K,J)=iym;  

  s_day(K,J)=ddliday;  

  e_day(K,J)=yint_d;  

end;  

  

/** 3.c If estimated weeks are known and currently not receiving, then count back from current

```

interview date or stop date if known to find the most recent year

the respondent could have begun receiving. Set start date as January of that year and count forward the number of months from January of that year and set

the stop date. Constrain start and stop dates by DLI and DCI, respectively. */

if wks(K,J)>=0 and cwks(K,J)=0 and yme(k,j)<=0 then do;

if 0<=mos(K,J)<=yint_m then do;

yms(K,J)=(yint_y*100)+01;

yme(K,J)=(yint_y*100)+01+mos(K,J);

end;

if (yint_m+01)<=mos(K,J)<=(yint_m+12) then do;

yms(K,J)=((yint_y-1)*100)+01;

if mos(K,J)<=12 then yme(K,J)=((yint_y-1)*100)+01+mos(K,J);

if mos(K,J)>=13 then yme(K,J)=(yint_y*100)+01+mos(K,J)-12;

end;

if (yint_m+13)<=mos(K,J)<=(yint_m+24) then do;

yms(K,J)=((yint_y-2)*100)+01;

if mos(K,J)<=24 then yme(K,J)=((yint_y-1)*100)+01+mos(K,J)-12;

if mos(K,J)>=25 then yme(K,J)=(yint_y*100)+01+mos(K,J)-24;

end;

if (yint_m+25)<=mos(K,J)<=(yint_m+36) then do;

yms(K,J)=((yint_y-3)*100)+01;

if mos(K,J)<=36 then yme(K,J)=((yint_y-1)*100)+01+mos(K,J)-24;

if mos(K,J)>=37 then yme(K,J)=(yint_y*100)+01+mos(K,J)-36;

end;

if (yint_m+37)<=mos(K,J)<=(yint_m+48) then do;

yms(K,J)=((yint_y-4)*100)+01;

if mos(K,J)<=48 then yme(K,J)=((yint_y-1)*100)+01+mos(K,J)-36;

if mos(K,J)>=49 then yme(K,J)=(yint_y*100)+01+mos(K,J)-48;

end;

s_day(K,J)=31;

do Z=1 to 12;

if (yme(K,J)-(round(yms(K,J),100)))=mon(Z) then e_day(K,J)=uiday(Z);

end;

if yme(k,j)>iym then do;

yme(k,j)=iym;

e_day(k,j)=de(k,j);

if ee(k,j)=1 then e_day(k,j)=9;

if ee(k,j)=2 then e_day(k,j)=19;

if ee(k,j)=3 then e_day(k,j)=28;

end;

end;

eflag(k,j)=3;

end;

if yme(k,j)>0 and yms(k,j)=. then do;

yms(k,j)=dliym;

```

s_day(k,j)=ddliday;
if j in (1,5) then eflag(k,j)=3;
else if j in (2,3,4,6,7,8) then eflag(k,j)=7;
end;

```

/** Note: If either stop month or year is ge -3 then it implies that this unemp spell has ended. Even if the respondent reports currently receiving that would imply another loop. **/

/** 4. If stop year is known and stop month is unknown. **/
 if yms(K,J)>0 and ye(K,J)>0 and -3<=me(K,J)<=-1 then do;

/** 4.a If estimated weeks are known, then count forward from start year. If the number of months falls short of the stop year, then set stop date to January of the

stop year. If the number of months exceeds the stop year, then set stop date to December of the stop year. Constrain stop date by DCI. If currently receiving then use DCI as stop date. **/
 if wks(K,J)>=0 then do;

Nms(k,j)=yms(k,j)- round (yms(k,j),100);

```

if (          0)<=mos(k,j)<=(12-Nms(k,j)+  0) then yme(k,j)=yms(k,j)+mos(k,j);
if (12-Nms(k,j)+  1)<=mos(k,j)<=(12-Nms(k,j)+ 12) then yme(k,j)=yms(k,j)+ 100+mos(k,j)-
12;
if (12-Nms(k,j)+ 13)<=mos(k,j)<=(12-Nms(k,j)+ 24) then yme(k,j)=yms(k,j)+ 200+mos(k,j)-
24;
if (12-Nms(k,j)+ 25)<=mos(k,j)<=(12-Nms(k,j)+ 36) then yme(k,j)=yms(k,j)+ 300+mos(k,j)-
36;
if (12-Nms(k,j)+ 37)<=mos(k,j)<=(12-Nms(k,j)+ 48) then yme(k,j)=yms(k,j)+ 400+mos(k,j)-
48;
if (12-Nms(k,j)+ 49)<=mos(k,j)<=(12-Nms(k,j)+ 60) then yme(k,j)=yms(k,j)+ 500+mos(k,j)-
60;
if (12-Nms(k,j)+ 61)<=mos(k,j)<=(12-Nms(k,j)+ 72) then yme(k,j)=yms(k,j)+ 600+mos(k,j)-
72;
if (12-Nms(k,j)+ 73)<=mos(k,j)<=(12-Nms(k,j)+ 84) then yme(k,j)=yms(k,j)+ 700+mos(k,j)-
84;
if (12-Nms(k,j)+ 85)<=mos(k,j)<=(12-Nms(k,j)+ 96) then yme(k,j)=yms(k,j)+ 800+mos(k,j)-
96;
if (12-Nms(k,j)+ 97)<=mos(k,j)<=(12-Nms(k,j)+108) then yme(k,j)=yms(k,j)+ 900+mos(k,j)-
108;
if (12-Nms(k,j)+109)<=mos(k,j)<=(12-Nms(k,j)+120) then yme(k,j)=yms(k,j)+1000+mos(k,j)-
120;
if (12-Nms(k,j)+121)<=mos(k,j)<=(12-Nms(k,j)+132) then yme(k,j)=yms(k,j)+1100+mos(k,j)-
132;

```

do Z=1 to 12;
 if yme(K,J)-round(yme(K,J),100)=mon(Z) then e_day(K,J)=uiday(Z);
 end;

if yme(K,J)>((ye(K,J)*100)+12) then do;
 yme(K,J)=((ye(K,J)*100)+12);
 e_day(K,J)=31;

```

end;

if yme(K,J)<((ye(K,J)*100)+01) then do;
  yme(K,J)=((ye(K,J)*100)+01);
  e_day(K,J)=31;
end;

if yme(K,J)>iym then do;
  yme(K,J)=iym;
  e_day(K,J)=yint_d;
end;
end;

/** 4.b If estimated weeks are unknown, then set stop date to December of stop year.
*/
if -3<=wks(K,J)<0 then do;
  yme(K,J)=((ye(K,J)*100)+12);
  e_day(K,J)=31;
end;

if yme(K,J)>iym then do;
  yme(K,J)=iym;
  e_day(K,J)=yint_d;
end;
if (eflag(k,j) not in (2,3,7) or eflag(k,j)=.) then eflag(k,j)=4;
if eflag(k,j) in (2,3,7) then eflag(k,j)=6;

end;

/** 5. If stop year is unknown.
*/
if -3<=ye(K,J)<=-1 then do;

/** 5.a If estimated weeks are known then count forward by the number of weeks from the start date
and constrain stop
date by DCI.
*/
if yms(K,J)>0 and wks(K,J)>=0 then do;

  Nms(k,j)=yms(k,j)- round (yms(k,j),100);

  if (          0)<=mos(k,j)<=(12-Nms(k,j)+  0) then yme(k,j)=yms(k,j)+mos(k,j);
  if (12-Nms(k,j)+  1)<=mos(k,j)<=(12-Nms(k,j)+ 12) then yme(k,j)=yms(k,j)+ 100+mos(k,j)-12;
  if (12-Nms(k,j)+ 13)<=mos(k,j)<=(12-Nms(k,j)+ 24) then yme(k,j)=yms(k,j)+ 200+mos(k,j)-24;
  if (12-Nms(k,j)+ 25)<=mos(k,j)<=(12-Nms(k,j)+ 36) then yme(k,j)=yms(k,j)+ 300+mos(k,j)-36;
  if (12-Nms(k,j)+ 37)<=mos(k,j)<=(12-Nms(k,j)+ 48) then yme(k,j)=yms(k,j)+ 400+mos(k,j)-48;
  if (12-Nms(k,j)+ 49)<=mos(k,j)<=(12-Nms(k,j)+ 60) then yme(k,j)=yms(k,j)+ 500+mos(k,j)-60;
  if (12-Nms(k,j)+ 61)<=mos(k,j)<=(12-Nms(k,j)+ 72) then yme(k,j)=yms(k,j)+ 600+mos(k,j)-72;

```

```

72;
  if (12-Nms(k,j)+ 73)<=mos(k,j)<=(12-Nms(k,j)+ 84) then yme(k,j)=yms(k,j)+ 700+mos(k,j)-
84;
  if (12-Nms(k,j)+ 85)<=mos(k,j)<=(12-Nms(k,j)+ 96) then yme(k,j)=yms(k,j)+ 800+mos(k,j)-
96;
  if (12-Nms(k,j)+ 97)<=mos(k,j)<=(12-Nms(k,j)+108) then yme(k,j)=yms(k,j)+ 900+mos(k,j)-
108;
  if (12-Nms(k,j)+109)<=mos(k,j)<=(12-Nms(k,j)+120) then yme(k,j)=yms(k,j)+1000+mos(k,j)-
120;
  if (12-Nms(k,j)+121)<=mos(k,j)<=(12-Nms(k,j)+132) then yme(k,j)=yms(k,j)+1100+mos(k,j)-
132;

```

```

    do Z=1 to 12;
      if yme(K,J)-round(yme(K,J),100)=mon(Z) then e_day(K,J)=uiday(Z);
    end;

```

```

    if yme(K,J)>iym then do;
      yme(K,J)=iym;
      e_day(K,J)=yint_d;
    end;
  end;

```

/** 5.b If estimated weeks are unknown then set stop date to December of the start year.

**/

```

    if yms(K,J)>0 and -3<=wks(K,J)<0 then do;
      yme(K,J)=(round (yms(K,J),100))+12;
      e_day(K,J)=31;
    end;

```

```

    if yme(K,J)>iym then do;
      yme(K,J)=iym;
      e_day(K,J)=yint_d;
    end;

```

```

      if (eflag(k,j) not in (2,3,7) or eflag(k,j)=.) then eflag(k,j)=5;
      if eflag(k,j) in (2,3,7) then eflag(k,j)=6;

```

end;

/** 6. If the start and stop years are unknown and estimated weeks are unknown then set start and stop dates to DLI and current interview, respectively. **/

```

    if -3<=ys(K,J)<=-1 and cwks(K,J)=0 and -3<=wks(K,J)<0 then do;

```

```

      yme(K,J)=iym;
      e_day(K,J)=yint_d;
      yms(K,J)=dliym;
      s_day(K,J)=ddliday;

```

```

      if j=1 then eflag(K,J)=6;
      else if j in (2,3,4,6,7,8) then eflag(K,J)=7;
    end;

```

```

/** This portion calculates the number of days receiving if start and end year and months are the
same **/
if (yms(k,j)=yme(k,j) and yms(k,j)>0 and e_day(k,j)>=0 and s_day(k,j)>=0) then do;
SAMEMONTH=1;
if (yms(k,j)-round(yms(k,j),100)) in (1,3,5,7,8,10,12) then do;
if e_day(k,j)>=(32-s_day(k,j)) then do;
s_day(k,j)=e_day(k,j)-(32-s_day(k,j))+1;
e_day(k,j)=0;
end;
else if e_day(k,j)<(32-s_day(k,j)) then do;
s_day(k,j)=0;
e_day(k,j)=0;
HANDEDIT=HANDEDIT+1;
end;
end;
if (yms(k,j)-round(yms(k,j),100)) in (4,6,9,11) then do;
if e_day(k,j)>=(31-s_day(k,j)) then do;
s_day(k,j)=e_day(k,j)-(31-s_day(k,j))+1;
e_day(k,j)=0;
end;
else if e_day(k,j)<(31-s_day(k,j)) then do;
s_day(k,j)=0;
e_day(k,j)=0;
HANDEDIT=HANDEDIT+1;
end;
end;
if (yms(k,j)-round(yms(k,j),100))=2 then do;
if e_day(k,j)>=(29-s_day(k,j)) then do;
s_day(k,j)=e_day(k,j)-(29-s_day(k,j))+1;
e_day(k,j)=0;
end;
else if e_day(k,j)<(29-s_day(k,j)) then do;
s_day(k,j)=0;
e_day(k,j)=0;
HANDEDIT=HANDEDIT+1;
end;
end;

```

/** This portion of the program uses the category reported by the respondent to create an estimated amount. The estimated amount is the midpoint rounded down. Note that the 12th category lists \$1251 as the amount. This amount was chosen since the category is unbounded - the number represents one dollar above the lower bound.**/

```

if -3<=inc(K,J)<=-1 and einc(K,J)=1 then incc(K,J)=(50/7);
if -3<=inc(K,J)<=-1 and einc(K,J)=2 then incc(K,J)=(125/7);
if -3<=inc(K,J)<=-1 and einc(K,J)=3 then incc(K,J)=(175/7);
if -3<=inc(K,J)<=-1 and einc(K,J)=4 then incc(K,J)=(225/7);
if -3<=inc(K,J)<=-1 and einc(K,J)=5 then incc(K,J)=(275/7);
if -3<=inc(K,J)<=-1 and einc(K,J)=6 then incc(K,J)=(351/7);
if inc(K,J)>-4 then aflag(K,J)=0;

```

** Create the start and stop dates in Continuous Month Scheme;

```
csm(K,J)=(round(yms(K,J),100)-198000)*.12+(yms(K,J)-round(yms(K,J),100));
cem(K,J)=(round(yme(K,J),100)-198000)*.12+(yme(K,J)-round(yme(K,J),100));
cdli=(round(dliym,100)-198000)*.12+(dliym-round(dliym,100))-1;

/** added to zero out people who back report. */
cdli=(round(dliym,100)-198000)*.12+(dliym-round(dliym,100))-1;

end;                                /** [E3] **/
end;                                /** [E2] **/
```

***** HANDEDIT - 1;

** The start and or end days are missing. So we have to make some assumptions about them. We assume

```
the start day to be the 1st of the month and the end day to be the last day of the month.;
if pubid=185 then do;
sdayu21=30;
edayu21=31;
end;
if pubid=7550 then do;
sdayu21=31;
end;
if pubid=4834 then do;
sdayu11=50;
edayu11=31;
end;
if pubid=8069 then do;
sdayu21=31;
edayu21=30;
end;
```

** Respondent reported receiving WIC, but doesn't know whether stopped for more than a month and so we

do not have any information on stop date. So we create another loop from the month after the reported stop month to DCI, and put eflag=7, and incc=-3.;

```
if pubid=4717 then do;
csmu11=dlicm; sdayu11=dliday; cemu11=doicm; edayu11=yint_d; eflagu11=7; aflagu11=0; inccu11=-
3;
end;
```

** Respondents reported receiving at DLI, currently not receiving and no periods of 1 month or more when they did not receive. So we take this to be receiving till less than a month before the interview.;

```
if pubid in (881,1198,3036,4613,4778,8099) then do;
cemu11=doicm-1; edayu11=yint_d; Neflagu11=1;
end;
```

** Respondents reported currently receiving, a period of 1 month or more when they did not receive, have a valid stop date but when asked if they received again, said NO so did not go through the second loop. We create another loop from the month after the reported stop month to DCI, and put eflag=7, incc=-3 and who=-3. However if the stop date is DCI, then we do not handedit.;

```

if pubid in (78,695,4611,6874) then do;
csmu12=cemu11+1;
if (ymeu11-(round(ymeu11, 100)))<12 then ymsu12=ymeu11+1;
if (ymeu11-(round(ymeu11, 100)))=12 then ymsu12=ymeu11-12+101;
do z=1 to 11;
if (ymeu11-(round(ymeu11, 100)))=mon(z) then sdayu12=uiday(z+1);
end;
if (ymeu11-(round(ymeu11, 100)))=12 then sdayu12=31;
cemu12=doicm; ymeu12=iym; edayu12=yint_d; eflagu12=7; aflagu12=0; inccu12=-3;
end;

```

***** END OF HANDEDIT - 1;

** Create the eflag variables for cases where they were not created as in the case where spell ended before DLI earlier.;

```

do k=1 to dim(csm,1);
do J=1 to dim(csm,2);
if eflag(k,j)=. and Neflag(k,j)~=. then do;
eflag(k,j)=Neflag(k,j);
end;
end;
end;

```

```

/** This portion counts the number of Unemployment Spells in each year          */
do K=1 to DIM(csm,1);
  do J=1 to DIM(csm,2);
    if ((csm(k,j)>dlicm) and (0<csm(k,j)<=cem(k,j)))
    or ((csm(k,j)=dlicm) and (dli(k,j)~=-1) and (0<csm(k,j)<=cem(k,j)))
    or ((0<csm(k,j)<dlicm) and (cem(k,j)>=dlicm) and (dli(k,j)~=-1)) then do;
      if 1<=csm(K,J)<= 12 then nu80=nu80+1;
      if 13<=csm(K,J)<= 24 then nu81=nu81+1;
      if 25<=csm(K,J)<= 36 then nu82=nu82+1;
      if 37<=csm(K,J)<= 48 then nu83=nu83+1;
      if 49<=csm(K,J)<= 60 then nu84=nu84+1;
      if 61<=csm(K,J)<= 72 then nu85=nu85+1;
      if 73<=csm(K,J)<= 84 then nu86=nu86+1;
      if 85<=csm(K,J)<= 96 then nu87=nu87+1;
      if 97<=csm(K,J)<=108 then nu88=nu88+1;
      if 109<=csm(K,J)<=120 then nu89=nu89+1;
      if 121<=csm(K,J)<=132 then nu90=nu90+1;
      if 133<=csm(K,J)<=144 then nu91=nu91+1;
      if 145<=csm(K,J)<=156 then nu92=nu92+1;
      if 157<=csm(K,J)<=168 then nu93=nu93+1;
      if 169<=csm(K,J)<=180 then nu94=nu94+1;

```

```

if 181<=csm(K,J)<=192 then nu95=nu95+1;
if 193<=csm(K,J)<=204 then nu96=nu96+1;
if 205<=csm(K,J)<=216 then nu97=nu97+1;
if 217<=csm(K,J)<=228 then nu98=nu98+1;
if 229<=csm(K,J)<=240 then nu99=nu99+1;
if 241<=csm(K,J)<=252 then nu00=nu00+1;
if 253<=csm(K,J)<=264 then nu01=nu01+1;
if 265<=csm(K,J)<=276 then nu02=nu02+1;
if 277<=csm(K,J)<=288 then nu03=nu03+1;
end;
end;
end;

do I=1 to DIM(nu);
  CURRnu(I)=nu(I);
end;

/** This portion creates the month receiving indicator (UI) and the amount receiving each month
(UIAMT) **/


do K=1 to DIM(csm,1);
  do J=1 to DIM(csm,2);
    C=0;
    do L=1 to DIM(ui);
      C=C+1;
      if 0<=L<=doicm then do;
        if csm(K,J)<=C<=cem(K,J) then do;
          if (C=csm(K,J) and inc(K,J)>=0) then do;
            ui(L)=eflag(K,J);
            if s_day(K,J) ~= . then uiamt(L)= round ((inc(K,J)*s_day(K,J)), 1);
          end;
        else if (C=cem(K,J) and incc(K,J)>=0) then do;
          ui(L)=eflag(K,J);
          if e_day(K,J) ~= . then uiamt(L)= round ((incc(K,J)*e_day(K,J)), 1);
        end;
      else if csm(K,J)<C and C<cem(K,J) and incc(K,J)>=0 then do;
        ui(L)=eflag(K,J);
        if C in (169,171,173,175,176,178,180,181,183,185,187,188,
          190,192,193,195,197,199,200,202,204,205,207,209,
          211,212,214,216,217,219,221,223,224,226,228,229,
          231,233,235,236,238,240,241,243,245,247,248,250,
          252,253,255,257,259,260,262,264,265,267,269,271,
          272,274,276,277,279,281,283,284,286,288) then do;
          uiamt(L)= round ((incc(K,J)*31),1);
        end;
      else if C in (172,174,177,179,184,186,189,191,196,198,201,203,
          208,210,213,215,220,222,225,227,232,234,237,239,
          244,246,249,251,256,258,261,263,268,270,273,275,
          280,282,285,287) then do;
    end;
  end;
end;

```

```

        uiamt(L)=round ((incc(K,J)*30),1);
end;
else if C in (194,242) then do;
    uiamt(L)=round ((incc(K,J)*29),1);
end;
else if C in (170,182,206,218,230,254,266,278) then do;
    uiamt(L)= round ((incc(K,J)*28),1);
end;
end;

else if incc(K,J)<0 then do;
    ui(L)=eflag(K,J);
    uiamt(L)=inc(K,J);
end;

if uiamt(L)<=10000 and (uiodd=0 OR uiodd=.) then do; uiodd=0; end;
if uiamt(L)>10000 then do; uiodd=uiamt(L); end;
if ui(L)>0 and uiamt(L)=. then do; uick=uick+1; end;
if (uiamt(L)>0 OR (-3<=uiamt(L)<0)) and ui(L)=0 then do; uickm=uickm+1; end;
if ui(L)=-3 and uiamt(L)~-=-3 then do; uickmm=uickmm+1; end;
if uiamt(L)=0 then dummyui=1;
end;
end;

end;
end;
end;
***** HANDEDIT - 2;
```

** This R reported first spell ending on October 10, 2002 and the second one starting from October 30, 2002. The program counted only the amount received for the month of October, 2002 from the second spell and overwrote the amount received from the first spell. In this round there was only one such case (i.e. where spells did not overlap but one ended and the next one started in the same month).
In future rounds such cases should be taken care of within the program itself.;
if pubid=2262 then do;
UIAMT274=(10*inccu21+1*inccu22);
end;

***** END OF HANDEDIT - 2;

/** This portion deletes spells that started before DLI

**/

```

do K=1 to DIM(csm,1);
  do J=1 to DIM(csm,2);
    C=0;
    do L=1 to DIM(ui);
      C=C+1;
      if csm(K,J)<=cdli and p010=1 and csm(K,J)~=. then do;
        if csm(K,J)<=C<=cdli then do;
          ui(L)=0;
          uiamt(L)=.;
```

```

        end;
        end;
        end;
    end;

END;                                     /** [E1] **/


/** This portion creates the months receiving in each year variables (UMXX) and the amount receiving
each year variables (UAMTXX).                                         */
do I=1 to DIM(um);
  if ui_ever>-4 OR p1210>-4 OR p1210=-5 then do;
    um(I)=0;
    uamt(I)=0;
    ua(I)=0;
    un(I)=0;
  end;
end;

if ui_ever>-4 OR p1210>-4 OR p1210=-5 then do;                         /** [B4] **/


/** calculate total no. months to divide the problematic amounts (too high) over. */
do L=1 to DIM(ui);
  if ui(L)>0 and uiamt(L)>10000 and uiodd>0 then do;
    uimnths=uimnths+1;
  end;
end;

/** divide the amount by the number of months to get an approximate "correct" monthly
amount.                                                       */
do L=1 to DIM(ui);
  if uiamt(L)>10000 and uimnths>0 and ui(L)>0 then do;
    uiamt(L)=round(uiamt(L)/uimnths, 1);
  end;
end;

** create the monthly receiving and amount variables.;

do i=1 to dim(UITOTAL);
do l=((i-1)*12)+1 to (I*12);
if UM(i)>=0 and UI(l)>0 then do;
  UM(i)=UM(i)+1;
end;
if UI(l)>1 then UM(i)=-3;
if -3<=UIAMT(l)<=-1 then do;
  un(i)=un(i)+1;
  UAMT(i)=UIAMT(l);
end;
if UIAMT(l)>=0 then do;
  ua(i)=ua(i)+1;
  if ua(i)=1 and un(i)=0 then UAMT(i)=UIAMT(l);
end;

```

```

if ua(i)>1 and un(i)=0 then UAMT(i)=UAMT(i)+UIAMT(l);
if UM(i)=-3 then UAMT(i)=-3;
end;
end;
end;

END; /** [E4] ***/

```

```

/** Part 3. Combine information from earlier rounds          */
/** Calculate total months ever received.                  ***/

```

```

do L=1 to DIM(ui);
  if ttlu>=0 and ui(L)>0 then do;
    ttlu=ttlu+1;
  end;
  if ui(L)>1 then ttlu=-3;
  if -3<=ui(L)<0 then ttlu=-3;
end;

```

```
totalu=TTLU;
```

```

if ttlu>=0 and ui_ever>=0 then do;
  ttlu=ttlu+ui_ever;
end;
if ttlu<0 OR ui_ever<0 then do;
  ttlu=-3;
end;

```

```
/** Calculate the number of unemployment spells in each year.
```

```

/**/
do I=1 to DIM(nu);
  if ui_spells(I)=-4 then ui_spells(I)=0;
  if nu(I)>=0 and ui_spells(I)>=0 then do;
    nu(I)=nu(I)+ui_spells(I);
  end;
  if (ui_spells(I)>=-3 and ui_spells(I)<=-1) then do;
    nu(I)=-3;
  end;
  if nu(I)=0 and oldui_spells(I)=-4 then do;
    nu(I)=-4;
  end;
end;

```

```

/** Calculate the total number of months and total amount receiving in each year.      */
do I=1 to DIM(uitotal);
  if uitotal(I)=-4 then uitotal(I)=0;
  if uiatotal(I)=-4 then uiatotal(I)=0;

  if um(I)>=0 and uitotal(I)>=0 then do;
    uitotal(I)=um(I)+uitotal(I);

```

```

end;
if um(I)<0 OR uitotal(I)<0 then do;
    uitotal(I)=-3;
end;
if uamt(I)>=0 and uiatotal(I)>=0 then do;
    uiatotal(I)=uamt(I)+uiatotal(I);
end;
if uamt(I)<0 OR uiatotal(I)<0 OR uitotal(I)<0 then do;
    uiatotal(I)=-3;
end;
if uamt(I)=0 and uiatotal(I)=-4 and uitotal(I)=0 then do;
    uiatotal(I)=-4;
end;
end;
end;

```

***** HANDEDIT - 3;

** these two Rs received UNEMP COMP at dli and in this round report that they stopped receiving a few days after dli. By convention we count UNEMP spells in months and so their spells would stop in the month of DLI. We add these amounts to the amounts that they reported receiving in the year of DLI. Only the created variable uiatotal is handedited. All other variables are kept as they are.;

```

if pubid=4559 then do;
UIATOTAL02=UIATOTAL02+407.1435;
end;
if pubid=4834 then do;
UIATOTAL02=UIATOTAL02+0;
end;
if pubid=7427 then do;
UIATOTAL01=UIATOTAL01+171.4284;
end;

```

***** END OF HANDEDIT - 3;

```

if p1210>-4 then do;
do k=1 to dim(csm,1);
do J=1 to dim(csm,2);
if 0<=cem(k,j)<dlicm then do;
cembfdli(k,j)=1;
cmd=1;
end;
end;
end;
end;

```

/** Assign Valid skips (-4) to those not eligible for program participation at DLI and -5 to noninterviews.**/

```

do I=1 to DIM(uitotal);
  if p1210=-4 then do;
    uitotal(I)=-4;
    nu(I)=-4;
    uiatotal(I)=-4;
    ttlu=-4;
  end;
  if p1210=-5 then do;
    uitotal(I)=-5;
    nu(I)=-5;
    uiatotal(I)=-5;
    ttlu=-5;
  end;
end;

/** To change all invalid skips to -3. */
array vars uitotal80-uitotal99 uitotal00 uitotal01 uitotal02 uitotal03
      uiatotal80-uiatotal99 uiatotal00 uiatotal01 uiatotal02 uiatotal03
      nu80-nu99 nu00 nu01 nu02 nu03 ttlu;
do over vars;
  if vars=-2 then vars=-3;
  if vars=-1 then vars=-3;
end;

```

***** SECTION 2: OTHER GOVERNMENT ASSISTANCE *****

Variables created: CV_GOVNT_PRG_EVER
 CV_GOVNT_PRG_YR.80-CV_GOVNT_PRG_YR.99 CV_GOVNT_PRG_YR.00
 CV_GOVNT_PRG_YR.01
 CV_GOVNT_PRG_YR.02 CV_GOVNT_PRG_YR.03
 CV_AMT_GOVNT_PCM_PCY.80-CV_AMT_GOVNT_PCM_PCY.99
 CV_AMT_GOVNT_PCM_PCY.00
 CV_AMT_GOVNT_PCM_PCY.01 CV_AMT_GOVNT_PCM_PCY.02
 CV_AMT_GOVNT_PCM_PCY.03

This program creates the variables that

- (1) indicate the number of months in any particular year (from 1980 to 2003) that the respondent received any of the four programs - AFDC, WIC, FOOD STAMPS and OTHER programs
 CV_GOVNT_PRG_YR.80-CV_GOVNT_PRG_YR.99, CV_GOVNT_PRG_YR.00,
 CV_GOVNT_PRG_YR.01, CV_GOVNT_PRG_YR.02 and CV_GOVNT_PRG_YR.03
- (2) indicate the TOTAL number of months that the respondent received any of the above four programs in a year (1980 to 2003) - CV_GOVNT_PRG_EVER
- (3) indicate the amount received in any particular year (from 1980 to 2003) by the respondent in any of the above programs CV_AMT_GOVNT_PCM_PCY.80-CV_AMT_GOVNT_PCM_PCY.99,
 CV_AMT_GOVNT_PCM_PCY.00, CV_AMT_GOVNT_PCM_PCY.01,
 CV_AMT_GOVNT_PCM_PCY.02 and CV_AMT_GOVNT_PCM_PCY.03

***** HAND EDITS (I) Based on comments;

** This respondent was receiving as of January 2002;
if pubid=4809 then do;
ps220001M=1;
end;

** This respondent reported that R was no longer living with recipient/s (partner and child) and so deleting all receiving information.;
if pubid=7725 then do;
ps21900A1=0; ps21430=0; ps233001=-4; ps234001=-4;
ps2350012=0; ps2350013=0; pS226001Y=-4; ps226001M=-4;
end;

***** HAND EDITS (II) Based on data and questionnaire problems;

** This Respondent reported two spells. In the first R reported receiving at DLI, never stopped for a month yet was not currently receiving. In the second spell R reported receiving from the month after DLI and stopped at current interview month. Since the amount received and the person receiving are the same in both spells we are putting the NO GAPS question to be 1. This will create two identical spells.;
if pubid=4249 then do;
ps359501=1;
end;

** These respondents reported currently receiving, never stopping for a month and yet were asked whether received again to which respondents below replied yes and so went through the next loop. Deleting this new spell information.;

** WIC;
if pubid=7958 then do;
p226001Y=-4; p220002M=-4; p222002=-4; p223002=-4; p233002=-4; p238002=-4;
p2350021=-4; p2350022=-4; p2350023=-4; p2350024=-4; p2350025=-4; p23500295=-4; p23500299=-4;
end;

** OTHER ASSISTANCE;
if pubid in (1808,1843,6489,7143,7525) then do;
p310002Y=-4; p310003Y=-4; p310004Y=-4;
p310002M=-4; p310003M=-4; p310004M=-4;
p312002=-4; p312003=-4; p312004=-4;
p313002=-4; p313003=-4; p313004=-4;
p323002=-4; p323003=-4; p323004=-4;
p3250021=-4; p3250031=-4; p3250041=-4;
p3250022=-4; p3250032=-4; p3250042=-4;
p3250023=-4; p3250033=-4; p3250043=-4;
p3250024=-4; p3250034=-4; p3250044=-4;
p3250025=-4; p3250035=-4; p3250045=-4;
p32500295=-4; p32500395=-4; p32500495=-4;
p32500299=-4; p32500399=-4; p32500499=-4;
end;

if pubid=2012 then do;
p310003Y=-4; p310003M=-4; p312003=-4; p313003=-4; p323003=-4;
p3250031=-4; p3250032=-4; p3250033=-4; p3250034=-4; p3250035=-4; p32500395=-4; p32500399=-4;

```
p310004Y=-4; p310004M=-4; p312004=-4; p313004=-4; p323004=-4;  
p3250041=-4; p3250042=-4; p3250043=-4; p3250044=-4; p3250045=-4; p32500495=-4; p32500499=-  
4;  
end;
```

```
** This respondent did not know whether CURRENTLY RECEIVING - FOR AFDC, WIC AND Food  
Stamps. Converting  
all responses these to NOT CURRENTLY RECEIVING.;  
if pubid=7718 then do;  
ps21397=0; ps35707=0; ps16410=0;  
end;
```

```
** This respondent should not have any program receiving information as R did not receive, someone  
else  
in the family did.;  
if pubid=7718 then do;  
ps16310=-4; ps16330=-4; ps16410=-4; ps19063A1=-4; ps197001=-4; ps201001=-4; ps203001=-4;  
ps204001=-4; ps207001=-4; ps208001=-4; ps2090011=-4; ps2090012=-4; ps2090013=-4; ps2090014=-  
4;  
ps2090015=-4; ps20900195=-4; ps20900199=-4; ps210001=-4; ps21100A1=-4; ps212001=-4;  
end;
```

```
***** END OF HAND EDITS (I) & (II) *****;
```

```
*** This part initializes the created variables at the value at date of last interview (DLI). It also creates  
date of birth, date of current interview (DCI) etc. variables in continuous month scheme (CMS) and  
YYYYMM scheme.
```

```
** Arrays of created variables;
```

```
array GTOTAL (24) GTOTAL80-GTOTAL99 GTOTAL00 GTOTAL01 GTOTAL02 GTOTAL03;  
array gatotal (24) GATOTAL80-GATOTAL99 GATOTAL00 GATOTAL01 GATOTAL02  
GATOTAL03;
```

```
** Arrays of other intermediate variables;
```

```
array ogtotal (24) ogtotal80-ogtotal99 ogtotal00 ogtotal01 ogtotal02 ogtotal 103;
```

```
array g_total (5,24) g5total80-g5total99 g5total00 g5total01 g5total02 g5total03;  
g4total80-g4total99 g4total00 g4total01 g4total02 g4total03;  
g3total80-g3total99 g3total00 g3total01 g3total02 g3total03;  
g2total80-g2total99 g2total00 g2total01 g2total02 g2total03;  
g1total80-g1total99 g1total00 g1total01 g1total02 g1total03;
```

```
array ogatotal (24) ogatotal80-ogatotal99 ogatotal00 ogatotal01 ogatotal02 ogatotal03;
```

```
array ga_total (5,24) ga5total80-ga5total99 ga5total00 ga5total01 ga5total02 ga5total03;  
ga4total80-ga4total99 ga4total00 ga4total01 ga4total02 ga4total03;  
ga3total80-ga3total99 ga3total00 ga3total01 ga3total02 ga3total03;  
ga2total80-ga2total99 ga2total00 ga2total01 ga2total02 ga2total03;  
ga1total80-ga1total99 ga1total00 ga1total01 ga1total02 ga1total03;
```

```

array gprgever (5) g5prg_ever g4prg_ever g3prg_ever g2prg_ever g1prg_ever;
array inty (5) int01_y int00_y int99_y int98_y int97_y;
array intm (5) int01_m int00_m int99_m int98_m int97_m;

** Identify the last round that respondent was interviewed (dli_rnd);
do i=1 to 5;
if inty(I)~=-5 then do;
dli_rnd=i;
i=5;
end;
end;

** Initialize to DLI values;

do i=1 to 5;
if dli_rnd=I then do;
gprg_ever=gprgever(i);
do j=1 to 24;
GTOTAL(j)=g_total(i,j);
ogtotal(j)=g_total(i,j);
GATOTAL(j)=ga_total(i,j);
ogatotal(j)=ga_total(i,j);
end;
end;
end;

/* Take care of cases in earlier rounds where although respondent did not participate in government
programs, the amount
received variable is 0 instead of -4. */
do j=1 to 24;
if GTOTAL(j)=0 and GATOTAL(j)=0 then ogatotal(j)=-4;
end;

** To facilitate computation convert missing gtotal values to 0. ;
do j=1 to 24;
if GTOTAL(j)=. then do;
ogtotal(j)=-4;
GTOTAL(j)=0;
end;
end;

** To facilitate computation convert missing gatotal values to 0. ;
do j=1 to 24;
if GATOTAL(j)=. then do;
ogatotal(j)=-4;
GATOTAL(j)=0;
end;
end;

```

```

** To facilitate computation convert -4 values to 0 for gprg_ever;
if gprg_ever=-4 then gprg_ever=0;

** Create various important date variables in the two schemes - the YYYYMM scheme and the
continuous
month scheme (CMS)
ymb - Date of Birth in YYYYMM scheme
cmb - Date of Birth in CMS
ym14 - Date of 14th birthday in YYYYMM scheme
cm14 - Date of 14th birthday in CMS
dliym - DLI in YYYYMM scheme
dlicm - DLI in CMS
iym - DCI in YYYYMM scheme
doicm - DCI in CMS
aaiym - Age at DCI in YYYYMM scheme
aaicm - Age at DCI in CMS;

ybm=(BDATE_Y*100)+BDATE_M;
cmb=((BDATE_Y-1980)*12)+BDATE_M;
ym14=((BDATE_Y+14)*100)+BDATE_M;
cm14=((BDATE_Y+14-1980)*12)+BDATE_M;

if p1210=1 then do;
do i=1 to 5;
if dli_rnd=I then do;
if intm(i)<12 then dliym=(inty(i)*100)+intm(i)+1;
if intm(i)=12 then dliym=((inty(i)+1)*100)+1;
dlicm=(inty(i)-1980)*12)+intm(i)+1;
end;
end;
end;
if p1210~=1 then dliym=ym14;
if p1210~=1 then dlicm=cm14;

if yint_y>0 and yint_m>0 then iym=(yint_y*100)+yint_m;
if yint_y>0 and yint_m>0 then doicm=((yint_y-1980)*12)+yint_m;
if yint_y>0 and yint_m>0 then aaiym=(yint_y-bdate_y)*100+yint_m-bdate_m;
if yint_y>0 and yint_m>0 then aaicm=(yint_y-bdate_y)*12+yint_m-bdate_m;

if yint_y in (-1,-2,-3) or yint_m in (-1,-2,-3) then do;
iym=-3;
doicm=-3;
aaiym=-3;
aaicm=-3;
end;

** Arrays of monthly variables indicating whether respondent (or any other household member)
received,
the amount they received and which household member/s received.;
```

```

array A (288) A001-A288; array AAMT (288) AAMT001-AAMT288; array AHHM (288)
AHHM001-AHHM288;
array W (288) W001-W288; array WAMT (288) WAMT001-WAMT288; array WHHM (288)
WHHM001-WHHM288;
array F (288) F001-F288; array FAMT (288) FAMT001-FAMT288; array FHFM (288)
FHFM001-FHFM288;
array O (288) O001-O288; array OAMT (288) OAMT001-OAMT288; array OHFM (288)
OHFM001-OHFM288;

```

```

** Define dummy variables to be used for checking the created (array) variables;
ack=0; ackm=0; ackh=0;
wck=0; wckm=0; wckh=0;
fck=0; fckm=0; fckh=0;
ock=0; ockm=0; ockh=0;

```

```

/*2. This part initializes the monthly variables and defines the arrays for those eligible for Program
Participation.*/
if p1210>-4 then do; /* [B1] */
/* Rs eligible for a program in round 6 */
if ps35707>-3 then p16200=ps35707; /* AFDC */
if ps21397>-3 then p21500=ps21397; /* WIC */
if ps16377>-3 then p18900=ps16377; /* Food Stamps */
if ps36087>-3 then p30500=ps36087; /* Other */

```

```

** Initialize values of the monthly variables to be created;
do l=cmb to doicm;
A(l)=0;
W(l)=0;
F(l)=0;
O(l)=0;
end;

```

```

/** START YEAR information */
array ys (4,8) ps358001Y ps358002Y ps358003Y ps358004Y p167001Y p167002Y p167003Y
p167004Y
                                ps220001Y ps220002Y ps220003Y ps220004Y p220001Y p220002Y p220003Y
p220004Y
                                ps191001Y ps191002Y ps191003Y ps191004Y p194001Y p194002Y p194003Y
p194004Y
                                ps310001Y ps310002Y ps310003Y ps310004Y p310001Y p310002Y p310003Y
p310004Y;

```

```

/** START MONTH information */
array ms (4,8) ps358001M ps358002M ps358003M ps358004M p167001M p167002M p167003M
p167004M
                                ps220001M ps220002M ps220003M ps220004M p220001M p220002M p220003M
p220004M
                                ps191001M ps191002M ps191003M ps191004M p194001M p194002M p194003M
p194004M
                                ps310001M ps310002M ps310003M ps310004M p310001M p310002M p310003M

```

```

p310004M;

/** end YEAR information */
array ye (4,8) ps358401Y ps358402Y ps358403Y ps358404Y p172001Y p172002Y p172003Y
p172004Y          ps226001Y ps226002Y ps226003Y ps226004Y p226001Y p226002Y p226003Y
p226004Y          ps200001Y ps200002Y ps200003Y ps200004Y p200001Y p200002Y p200003Y
p200004Y          ps316001Y ps316002Y ps316003Y ps316004Y p316001Y p316002Y p316003Y
p316004Y;

/** end MONTH information */
array me (4,8) ps358401M ps358402M ps358403M ps358404M p172001M p172002M p172003M
p172004M          ps226001M ps226002M ps226003M ps226004M p226001M p226002M p226003M
p226004M          ps200001M ps200002M ps200003M ps200004M p200001M p200002M p200003M
p200004M          ps316001M ps316002M ps316003M ps316004M p316001M p316002M p316003M
p316004M;

** START DATE in YYYYMM scheme;
array yms (4,8) ymsa11-ymsa14 ymsa21-ymsa24
               ymsw11-ymsw14 ymsw21-ymsw24
               ymsf11-ymsf14 ymsf21-ymsf24
               ymso11-ymso14 ymso21-ymso24;

** START DATE in CMS;
array csm (4,8) csma11-csma14 csma21-csma24
               csmw11-csmw14 csmw21-csmw24
               csmf11-csmf14 csmf21-csmf24
               csmo11-csmo14 csmo21-csmo24;

** STOP DATE in YYYYMM scheme;
array yme (4,8) ymea11-yme14 ymea21-yme24
               ymew11-ymew14 ymew21-ymew24
               ymef11-ymef14 ymef21-ymef24
               ymeo11-ymeo14 ymeo21-ymeo24;

** STOP DATE in CMS;
array cem (4,8) cema11-cema14 cema21-cema24
               cemw11-cemw14 cemw21-cemw24
               cemf11-cemf14 cemf21-cemf24
               cemo11-cemo14 cemo21-cemo24;

/** CURRENTLY RECEIVING information */
array cur (4,8) ps359501 ps359502 ps359503 ps359504 p169001 p169002 p169003 p169004
               ps23700A1 ps23700A2 ps23700A3 ps23700A4 p222001 p222002 p222003 p222004
               ps21100A1 ps21100A2 ps21100A3 ps21100A4 p196001 p196002 p196003 p196004
               ps32700A1 ps32700A2 ps32700A3 ps32700A4 p312001 p312002 p312003 p312004;

```

```

array cwks (4,8) ps35740 ps35740 ps35740 ps35740 p16200 p16200 p16200 p16200
          ps21430 ps21430 ps21430 ps21430 p21500 p21500 p21500 p21500
          ps16410 ps16410 ps16410 ps16410 p18900 p18900 p18900 p18900
          ps36120 ps36120 ps36120 ps36120 p30500 p30500 p30500 p30500;

** RECEIVING at DLI, chk;
array dli (4,8) ps357901 ps357902 ps357903 ps357904 DUMa1 DUMa2 DUMa3 DUMa4
          ps21900A1 ps21900A2 ps21900A3 ps21900A4 DUMw1 DUMw2 DUMw3 DUMw4
          ps19063A1 ps19063A2 ps19063A3 ps19063A4 DUMf1 DUMf2 DUMf3 DUMf4
          ps30900A1 ps30900A2 ps30900A3 ps30900A4 DUMo1 DUMo2 DUMo3 DUMo4;

** AMOUNT RECEIVED - ACTUAL;
array inc (4,8) ps359001 ps359002 ps359003 ps359004 p181001 p181002 p181003 p181004
          ps233001 ps233002 ps233003 ps233004 p233001 p233002 p233003 p233004
          ps207001 ps207002 ps207003 ps207004 p207001 p207002 p207003 p207004
          ps323001 ps323002 ps323003 ps323004 p323001 p323002 p323003 p323004;

** AMOUNT RECEIVED - ESTIMATED;
array einc (4,8) ps359101 ps359102 ps359103 ps359104 p182001 p182002 p182003 p182004
          ps234001 ps234002 ps234003 ps234004 p234001 p234002 p234003 p234004
          ps208001 ps208002 ps208003 ps208004 p208001 p208002 p208003 p208004
          ps324001 ps324002 ps324003 ps324004 p324001 p324002 p324003 p324004;

** AMOUNT RECEIVED - COMPUTED from AMOUNT RECEIVED, ACTUAL and ESTIMATED;
array incc (4,8) incca11-incca14 incca21-incca24
          inccw11-inccw14 inccw21-inccw24
          inccf11-inccf14 inccf21-inccf24
          incco11-incco14 incco21-incco24;

** ESTIMATED WEEKS - unable to answer start or stop quex;
array wks (4,8) ps358601 ps358602 ps358603 ps358604 p175001 p175002 p175003 p175004
          ps228001 ps228002 ps228003 ps228004 p228001 p228002 p228003 p228004
          ps202001 ps202002 ps202003 ps202004 p202001 p202002 p202003 p202004
          ps318001 ps318002 ps318003 ps318004 p318001 p318002 p318003 p318004;

** ESTIMATED MONTHS - computed from ESTIMATED WEEKS - unable to answer start or stop
quex;
array mos (4,8) mosa11-mosa14 mosa21-mosa24
          mosw11-mosw14 mosw21-mosw24
          mosf11-mosf14 mosf21-mosf24
          moso11-moso14 moso21-moso24;

** PERSON RECEIVING - RESPONDENT;
array rcvr (4,8) ps3592011 ps3592021 ps3592031 ps3592041 p1830011 p1830021 p1830031
          p1830041
          ps2350011 ps2350021 ps2350031 ps2350041 p2350011 p2350021 p2350031
          p2350041
          ps2090011 ps2090021 ps2090031 ps2090041 p2090011 p2090021 p2090031
          p2090041
          ps3250011 ps3250021 ps3250031 ps3250041 p3250011 p3250021 p3250031
          p3250041

```

p3250041;

** PERSON RECEIVING - SPOUSE;

array rcvs (4,8) ps3592012 ps3592022 ps3592032 ps3592042 p1830012 p1830022 p1830032
p1830042
p2350042
p2090042
p3250042;

ps2350012 ps2350022 ps2350032 ps2350042 p2350012 p2350022 p2350032
ps2090012 ps2090022 ps2090032 ps2090042 p2090012 p2090022 p2090032
ps3250012 ps3250022 ps3250032 ps3250042 p3250012 p3250022 p3250032

** PERSON RECEIVING - CHILD;

array rcvc (4,8) ps3592013 ps3592023 ps3592033 ps3592043 p1830013 p1830023 p1830033
p1830043
p2350043
p2090043
p3250043;

ps2350013 ps2350023 ps2350033 ps2350043 p2350013 p2350023 p2350033
ps2090013 ps2090023 ps2090033 ps2090043 p2090013 p2090023 p2090033
ps3250013 ps3250023 ps3250033 ps3250043 p3250013 p3250023 p3250033

** PERSON RECEIVING - FAMILY MEMBER;

array rcvf (4,8) ps3592014 ps3592024 ps3592034 ps3592044 p1830014 p1830024 p1830034
p1830044
p2350044
p2090044
p3250044;

ps2350014 ps2350024 ps2350034 ps2350044 p2350014 p2350024 p2350034
ps2090014 ps2090024 ps2090034 ps2090044 p2090014 p2090024 p2090034
ps3250014 ps3250024 ps3250034 ps3250044 p3250014 p3250024 p3250034

** PERSON RECEIVING - OTHER PERSON;

array rcvo (4,8) ps3592015 ps3592025 ps3592035 ps3592045 p1830015 p1830025 p1830035
p1830045
p2350045
p2090045
p3250045;

ps2350015 ps2350025 ps2350035 ps2350045 p2350015 p2350025 p2350035
ps2090015 ps2090025 ps2090035 ps2090045 p2090015 p2090025 p2090035
ps3250015 ps3250025 ps3250035 ps3250045 p3250015 p3250025 p3250035

** PERSON RECEIVING DUMMY;

array who (4,8) whoa11-whoa14 whoa21-whoa24
whow11-whow14 whow21-whow24
whof11-whof14 whof21-whof24
whoo11-whoo14 whoo21-whoo24;

** EDIT FLAGS for DATES;

array eflag (4,8) eflaga11-eflaga14 eflaga21-eflaga24
eflagw11-eflagw14 eflagw21-eflagw24

```

eflagf11-eflagf14 eflagf21-eflagf24
eflago11-eflago14 eflago21-eflago24;

** EDIT FLAGS for DATES (when stop date is before DLI);
array Neflag (4,8) Nefлага11-Nefлага14 Nefлага21-Nefлага24
               Neflagw11-Neflagw14 Neflagw21-Neflagw24
               Neflagf11-Neflagf14 Neflagf21-Neflagf24
               Neflago11-Neflago14 Neflago21-Neflago24;

** EDIT FLAGS for AMOUNTS;
array aflag (4,8) aflaga11-aflaga14 aflaga21-aflaga24
               aflagw11-aflagw14 aflagw21-aflagw24
               aflagf11-aflagf14 aflagf21-aflagf24
               aflago11-aflago14 aflago21-aflago24;

** AGE FLAG;
array agefl (4,8) agefla11-agefla14 agefla21-agefla24
               ageflw11-ageflw14 ageflw21-ageflw24
               ageflf11-ageflf14 ageflf21-ageflf24
               ageflo11-ageflo14 ageflo21-ageflo24;

** DUMMY for STOP DATES before DLI;
array cembfdli (4,8) cembfdlia11-cembfdlia14 cembfdlia21-cembfdlia24
               cembfdliw11-cembfdliw14 cembfdliw21-cembfdliw24
               cembfdlif11-cembfdlif14 cembfdlif21-cembfdlif24
               cembfdlio11-cembfdlio14 cembfdlio21-cembfdlio24;

** Identifies the starting month - required in an intermediate step;
array nms (4,8) nmsa11-nmsa14 nmsa21-nmsa24
               nmsw11-nmsw14 nmsw21-nmsw24
               nmsf11-nmsf14 nmsf21-nmsf24
               nmso11-nmso14 nmso21-nmso24;

** To be used for checking eflags;
array efl (4,8) efla11-efla14 efla21-efla24
               eflw11-eflw14 eflw21-eflw24
               eflf11-eflf14 eflf21-eflf24
               eflo11-eflo14 eflo21-eflo24;

/** This portion of the SAS program defines the start date and stop dates. If the respondent reports still receiving, the DCI is used as the temporary end date for the last loop reported. In the next survey round, the respondent will be asked if he or she is still receiving and, if not, a permanent stop date equivalent to the interview date of the previous round will be assigned. Users will be able to tell which method was used by looking at the following participation flag variable created during the program. The categories are the following:
1=respondent reported participation dates
2=start month imputed
3=start month and year imputed
4=stop month imputed
5=stop month and year imputed
6=start and stop dates imputed
```

7=error in data due to round 5 questionnaire problem or inconsistent response. Both results in our not sure whether the respondents received in these months or not.

Questionnaire Error: These respondents reported valid start and stop dates but were never asked the follow-up question - if they received again.

Conflicting Response: These respondents reported currently receiving, a valid start date and a valid stop date that is different from the DCI but in the follow-up question as to whether they received again, they answered NO. */

```
do i=1 to 4;                                /** [b2] */
  do j=1 to 8;                            /** [b3] */

***** CREATE START and STOP DATES OF WELFARE SPELLS - FORMAT YYYYMM;
```

```
*** 1. set start date if both start month and year are valid;
```

```
  if ys(i,j)>0 and ms(i,j)>0 then yms(i,j)=(ys(i,j)*100)+ms(i,j);
```

```
** set start date to DLI if receiving then;
```

```
  if dli(i,j)=1 then do;
```

```
    yms(i,j)=dliym;
```

```
    if cwks(i,j)=1 then do;
```

```
      yme(i,j)=iym;
```

```
      NOGAPS=1;
```

```
    end;
```

```
  end;
```

```
** set stop date to DCI if receiving now;
```

```
  if cur(i,j)=1 then yme(i,j)=iym;
```

```
** set stop date if both end month and year are valid;
```

```
  if ye(i,j)>0 and me>0 then yme(i,j)=(ye(i,j)*100)+me(i,j);
```

```
** if end month greater than DCI, the constrain stop date by DCI;
```

```
  if yme(i,j)>iym then yme(i,j)=iym;
```

```
** eflag is not set if the stop date is before DLI as we do not compute months receiving variable in these cases. (eflag variables are used later to compute the months receiving variables). We create another eflag variable (Neflag) to know the eflag value;
```

```
  if yms(i,j)>0 and yme(i,j)>=dliym then eflag(i,j)=1;
```

```
  if yms(i,j)>0 and 0<yme(i,j)<dliym then Neflag(i,j)=1;
```

```
** compute estimated months received from estimated weeks received variable;
```

```
  if wks(i,j)>=0 and (ys(i,j)<0 OR ms(i,j)<0 OR ye(i,j)<0 OR me(i,j)<0) then do;
```

```
    mos(i,j)=floor(wks(i,j)/4.3);          /* determine months from estimated
```

```
weeks variable */
```

```
  end;
```

```
agefl(i,j)=0;
```

```
  if ys(i,j)>0 and ys(i,j)<1990 then agefl(i,j)=1;
```

```
/** 2. If start year is known and month is unknown **/
```

if ys(i,j)>0 and -3<=ms(i,j)<=-1 then do;

/** If estimated weeks are known and respondent is currently receiving, then count backwards by the number of weeks from DCI. If the number of estimated weeks falls short of the start year, then set start month to December of the start year. If the number of weeks is past the start year, then set start month to January of the start year. **/

```
if (wks(i,j)>=0 and cwks(i,j)=1) then do;
  yme(i,j)=iym;
  if (      0)<=mos(i,j)<=(yint_m- 1) then yms(i,j)=iym-mos(i,j);
  if (yint_m+ 0)<=mos(i,j)<=(yint_m+ 11) then yms(i,j)=iym- 100-(mos(i,j)- 12);
  if (yint_m+ 12)<=mos(i,j)<=(yint_m+ 23) then yms(i,j)=iym- 200-(mos(i,j)- 24);
  if (yint_m+ 24)<=mos(i,j)<=(yint_m+ 35) then yms(i,j)=iym- 300-(mos(i,j)- 36)
  if (yint_m+ 36)<=mos(i,j)<=(yint_m+ 47) then yms(i,j)=iym- 400-(mos(i,j)- 48);
  if (yint_m+ 48)<=mos(i,j)<=(yint_m+ 59) then yms(i,j)=iym- 500-(mos(i,j)- 60);
  if (yint_m+ 60)<=mos(i,j)<=(yint_m+ 71) then yms(i,j)=iym- 600-(mos(i,j)- 72);
  if (yint_m+ 72)<=mos(i,j)<=(yint_m+ 83) then yms(i,j)=iym- 700-(mos(i,j)- 84);
  if (yint_m+ 84)<=mos(i,j)<=(yint_m+ 95) then yms(i,j)=iym- 800-(mos(i,j)- 96);
  if (yint_m+ 96)<=mos(i,j)<=(yint_m+107) then yms(i,j)=iym- 900-(mos(i,j)-108);
  if (yint_m+108)<=mos(i,j)<=(yint_m+119) then yms(i,j)=iym-1000-(mos(i,j)-120);
  if (yint_m+120)<=mos(i,j)<=(yint_m+131) then yms(i,j)=iym-1100-(mos(i,j)-132);
  if yms(i,j)<((ys(i,j)*100)+01) then yms(i,j)=((ys(i,j)*100)+01);
  if yms(i,j)<dliym then yms(i,j)=dliym;
  if yms(i,j)>((ys(i,j)*100)+12) then yms(i,j)=((ys(i,j)*100)+12);
end;
```

/** If estimated weeks are missing and respondent is currently receiving then set stop date to DCI and start

date to December of the start year **/

```
else if (-3<=wks(i,j)<0 and cwks(i,j)=1) then do;
  yme(i,j)=iym;
  yms(i,j)=((ys(i,j)*100)+12);
  if yms(i,j)>iym then yms(i,j)=iym;
end;
```

/** If estimated weeks are missing and respondent is not currently receiving then set stop date to December and start date to

January of start year. Constrain stop date by DCI .**/

```
else if (-3<=wks(i,j)<0 and cwks(i,j)=0) then do;
  yms(i,j)=((ys(i,j)*100)+01);
  if yms(i,j)<dliym then yms(i,j)=dliym;
  if yms(i,j)>((ys(i,j)*100)+12) then yms(i,j)=((ys(i,j)*100)+12);
  if yme(i,j)=. then yme(i,j)=((ys(i,j)*100)+12);
  if yme(i,j)>iym then yme(i,j)=iym;
end;
```

/** If estimated weeks are known and respondent is not currently receiving, then set start date to January of start year. Constrain start date by DLI. But finally constrain by December of start year. Then count forward by the number of weeks from start date. Constrain stop date by DCI. **/

```

else if (wks(i,j)>=0 and cwks(i,j)=0) then do;
  yms(i,j)=((ys(i,j)*100)+01);
  if yms(i,j)<dliym then yms(i,j)=dliym;
  Nms(i,j)=yms(i,j)- round(yms(i,j),100);
  if (           0)<=mos(i,j)<=(12-Nms(i,j)+  0) then yme(i,j)=yms(i,j)+mos(i,j);
  if (12-Nms(i,j)+  1)<=mos(i,j)<=(12-Nms(i,j)+ 12) then yme(i,j)=yms(i,j)+ 100+mos(i,j)-
12;
  if (12-Nms(i,j)+ 13)<=mos(i,j)<=(12-Nms(i,j)+ 24) then yme(i,j)=yms(i,j)+ 200+mos(i,j)-
24;
  if (12-Nms(i,j)+ 25)<=mos(i,j)<=(12-Nms(i,j)+ 36) then yme(i,j)=yms(i,j)+ 300+mos(i,j)-
36;
  if (12-Nms(i,j)+ 37)<=mos(i,j)<=(12-Nms(i,j)+ 48) then yme(i,j)=yms(i,j)+ 400+mos(i,j)-
48;
  if (12-Nms(i,j)+ 49)<=mos(i,j)<=(12-Nms(i,j)+ 60) then yme(i,j)=yms(i,j)+ 500+mos(i,j)-
60;
  if (12-Nms(i,j)+ 61)<=mos(i,j)<=(12-Nms(i,j)+ 72) then yme(i,j)=yms(i,j)+ 600+mos(i,j)-
72;
  if (12-Nms(i,j)+ 73)<=mos(i,j)<=(12-Nms(i,j)+ 84) then yme(i,j)=yms(i,j)+ 700+mos(i,j)-
84;
  if (12-Nms(i,j)+ 85)<=mos(i,j)<=(12-Nms(i,j)+ 96) then yme(i,j)=yms(i,j)+ 800+mos(i,j)-
96;
  if (12-Nms(i,j)+ 97)<=mos(i,j)<=(12-Nms(i,j)+108) then yme(i,j)=yms(i,j)+ 900+mos(i,j)-
108;
  if (12-Nms(i,j)+109)<=mos(i,j)<=(12-Nms(i,j)+120) then yme(i,j)=yms(i,j)+1000+mos(i,j)-
120;
  if (12-Nms(i,j)+121)<=mos(i,j)<=(12-Nms(i,j)+132) then yme(i,j)=yms(i,j)+1100+mos(i,j)-
132;
  if yme(i,j)>iym then yme(i,j)=iym;
end;

```

** If stop date is known (then estimated weeks is not asked and so wks=-4) then set start date to January of the start year;

```

if yme(i,j)>0 and wks(i,j)=-4 then yms(i,j)=((ys(i,j)*100)+01);
if yms(i,j)<dliym then yms(i,j)=dliym;
if yms(i,j)>((ys(i,j)*100)+12) then yms(i,j)=((ys(i,j)*100)+12);

```

** eflag variables::

```

eflag(i,j)=2;

```

```

end;

```

*** 3. If start year is unknown;

```

if -3<=ys(i,j)<=-1 then do;

```

** If estimated weeks are known and respondent is currently receiving then count back from the DCI.;

```

if wks(i,j)>=0 and cwks(i,j)=1 then do;
  yme(i,j)=iym;
  if (      0)<=mos(i,j)<=(yint_m- 1) then yms(i,j)=iym-mos(i,j);
  if (yint_m+ 0)<=mos(i,j)<=(yint_m+ 11) then yms(i,j)=iym- 100-(mos(i,j)- 12);
  if (yint_m+ 12)<=mos(i,j)<=(yint_m+ 23) then yms(i,j)=iym- 200-(mos(i,j)- 24);
  if (yint_m+ 24)<=mos(i,j)<=(yint_m+ 35) then yms(i,j)=iym- 300-(mos(i,j)- 36);
  if (yint_m+ 36)<=mos(i,j)<=(yint_m+ 47) then yms(i,j)=iym- 400-(mos(i,j)- 48);
  if (yint_m+ 48)<=mos(i,j)<=(yint_m+ 59) then yms(i,j)=iym- 500-(mos(i,j)- 60);
  if (yint_m+ 60)<=mos(i,j)<=(yint_m+ 71) then yms(i,j)=iym- 600-(mos(i,j)- 72);
  if (yint_m+ 72)<=mos(i,j)<=(yint_m+ 83) then yms(i,j)=iym- 700-(mos(i,j)- 84);
  if (yint_m+ 84)<=mos(i,j)<=(yint_m+ 95) then yms(i,j)=iym- 800-(mos(i,j)- 96);
  if (yint_m+ 96)<=mos(i,j)<=(yint_m+107) then yms(i,j)=iym- 900-(mos(i,j)-108);
  if (yint_m+108)<=mos(i,j)<=(yint_m+119) then yms(i,j)=iym-1000-(mos(i,j)-120);
  if (yint_m+120)<=mos(i,j)<=(yint_m+131) then yms(i,j)=iym-1100-(mos(i,j)-132);
  if yms(i,j)<dliym then yms(i,j)=dliym;
end;

```

** If estimated weeks are missing and respondent is currently receiving then set start date to DLI and stop date to DCI;

```

if -3<=wks(i,j)<0 and cwks(i,j)=1 then do;
  yms(i,j)=dliym;
  yme(i,j)=iym;
end;

```

** If weeks are known and respondent is not currently receiving then count backward from DCI/or stop date if known and set start month to January of that year. Then count forward from this start date to get the stop date. Constrain stop date by DCI;

```

if wks(i,j)>=0 and cwks(i,j)=0 and yme(i,j)<=0 then do;
  if 0<=mos(i,j)<=yint_m then do;
    yms(i,j)=(yint_y*100)+01;
    yme(i,j)=(yint_y*100)+01+mos(i,j);
  end;
  if (yint_m+01)<=mos(i,j)<=(yint_m+12) then do;
    yms(i,j)=((yint_y-1)*100)+01;
    if 0<=mos(i,j)<=12 then yme(i,j)=((yint_y-1)*100)+01+mos(i,j);
    if 13<=mos(i,j) then yme(i,j)=(yint_y*100)+01+mos(i,j)-12;
  end;
  if (yint_m+13)<=mos(i,j)<=(yint_m+24) then do;
    yms(i,j)=((yint_y-2)*100)+01;
    if mos(i,j)<=24 then yme(i,j)=((yint_y-1)*100)+01+mos(i,j)-12;
    if 25<=mos(i,j) then yme(i,j)=(yint_y*100)+01+mos(i,j)-24;
  end;
  if (yint_m+25)<=mos(i,j)<=(yint_m+36) then do;
    yms(i,j)=((yint_y-3)*100)+01;
    if mos(i,j)<=36 then yme(i,j)=((yint_y-1)*100)+01+mos(i,j)-36;
    if 37<=mos(i,j) then yme(i,j)=(yint_y*100)+01+mos(i,j)-36;
  end;

```

```

end;
else if wks(i,j)>=0 and cwks(i,j)=0 and yme(i,j)>0 then do;
  if 0<=mos(i,j)<=me(i,j) then do;
    yms(i,j)=(ye(i,j)*100)+01;
    yme(i,j)=(ye(i,j)*100)+01+mos(i,j);
  end;
  if (me(i,j)+01)<=mos(i,j)<=(me(i,j)+12) then do;
    yms(i,j)=((ye(i,j)-1)*100)+01;
    if 0<=mos(i,j)<=12 then yme(i,j)=((ye(i,j)-1)*100)+01+mos(i,j);
    if 13<=mos(i,j) then yme(i,j)=(ye(i,j)*100)+01+mos(i,j)-12;
  end;
  if (me(i,j)+13)<=mos(i,j)<=(me(i,j)+24) then do;
    yms(i,j)=((ye(i,j)-2)*100)+01;
    if mos(i,j)<=24 then yme(i,j)=((ye(i,j)-1)*100)+01+mos(i,j)-12;
    if 25<=mos(i,j) then yme(i,j)=(ye(i,j)*100)+01+mos(i,j)-24;
  end;
  if (me(i,j)+25)<=mos(i,j)<=(me(i,j)+36) then do;
    yms(i,j)=((ye(i,j)-3)*100)+01;
    if mos(i,j)<=36 then yme(i,j)=((ye(i,j)-1)*100)+01+mos(i,j)-36;
    if 37<=mos(i,j) then yme(i,j)=(ye(i,j)*100)+01+mos(i,j)-36;
  end;
  if yme(i,j)>iym then yme(i,j)=iym;
end;

```

** eflag variables::

```
eflag(i,j)=3;
```

```
end;
```

** If stop date is known but start date is not, then set DLI as start date;

```

if yme(i,j)>0 and yms(i,j)=. then do;
  yms(i,j)=dliy;
  if j in (1,5) then eflag(i,j)=3;
  else if j in (2,3,4,6,7,8) then eflag(i,j)=7;
end;

```

*** 4. If stop year is known but stop month is not known;

```
if yms(i,j)>0 and ye(i,j)>0 and -2<=me(i,j)<=-1 then do;
```

** If estimated weeks are known, then count forward from start date. If the number of months falls short of the stop year, then set stop date to January of the stop year. If the number of months exceeds the stop year, then use December of the stop year as the stop date and constrain stop date by DCI;

```

if wks(i,j)>=0 then do;
  Nms(i,j)=yms(i,j)- round(yms(i,j),100);

```

```

if ( 0)<=mos(i,j)<=(12-Nms(i,j)+ 0) then yme(i,j)=yms(i,j)+ mos(i,j);
if (12-Nms(i,j)+ 1)<=mos(i,j)<=(12-Nms(i,j)+ 12) then yme(i,j)=yms(i,j)+ 100+mos(i,j)-
12;
if (12-Nms(i,j)+ 13)<=mos(i,j)<=(12-Nms(i,j)+ 24) then yme(i,j)=yms(i,j)+ 200+mos(i,j)-
24;
if (12-Nms(i,j)+ 25)<=mos(i,j)<=(12-Nms(i,j)+ 36) then yme(i,j)=yms(i,j)+ 300+mos(i,j)-
36;
if (12-Nms(i,j)+ 37)<=mos(i,j)<=(12-Nms(i,j)+ 48) then yme(i,j)=yms(i,j)+ 400+mos(i,j)-
48;
if (12-Nms(i,j)+ 49)<=mos(i,j)<=(12-Nms(i,j)+ 60) then yme(i,j)=yms(i,j)+ 500+mos(i,j)-
60;
if (12-Nms(i,j)+ 61)<=mos(i,j)<=(12-Nms(i,j)+ 72) then yme(i,j)=yms(i,j)+ 600+mos(i,j)-
72;
if (12-Nms(i,j)+ 73)<=mos(i,j)<=(12-Nms(i,j)+ 84) then yme(i,j)=yms(i,j)+ 700+mos(i,j)-
84;
if (12-Nms(i,j)+ 85)<=mos(i,j)<=(12-Nms(i,j)+ 96) then yme(i,j)=yms(i,j)+ 800+mos(i,j)-
96;
if (12-Nms(i,j)+ 97)<=mos(i,j)<=(12-Nms(i,j)+108) then yme(i,j)=yms(i,j)+ 900+mos(i,j)-
108;
if (12-Nms(i,j)+109)<=mos(i,j)<=(12-Nms(i,j)+120) then yme(i,j)=yms(i,j)+1000+mos(i,j)-
120;
if (12-Nms(i,j)+121)<=mos(i,j)<=(12-Nms(i,j)+132) then yme(i,j)=yms(i,j)+1100+mos(i,j)-
132;
if yme(i,j)>((ye(i,j)*100)+12) then yme(i,j)=((ye(i,j)*100)+12);
if yme(i,j)<((ye(i,j)*100)+01) then yme(i,j)=((ye(i,j)*100)+01);
if yme(i,j)>iym then yme(i,j)=iym;
end;

```

** If estimated weeks are missing then set stop date to December of stop year and constrain by DCI;

```

if -3<=wks(i,j)<0 then yme(i,j)=((ye(i,j)*100)+12);
if yme(i,j)>iym then yme(i,j)=iym;

```

** eflag variables::

```

if (eflag(i,j) not in (2,3,7) or eflag(i,j)=.) then eflag(i,j)=4;
if eflag(i,j) in (2,3,7) then eflag(i,j)=6;

```

end;

*** 5. If stop year is unknown;

```

if -3<=ye(i,j)<=-1 then do;

```

** If estimated weeks are known then count forward from start date;

```

if yms(i,j)>0 and wks(i,j)>=0 then do;
Nms(i,j)=yms(i,j)- round(yms(i,j),100);

```

```

if ( 0)<=mos(i,j)<=(12-Nms(i,j)+ 0) then yme(i,j)=yms(i,j)+ mos(i,j);
if (12-Nms(i,j)+ 1)<=mos(i,j)<=(12-Nms(i,j)+ 12) then yme(i,j)=yms(i,j)+ 100+mos(i,j)- 12;
if (12-Nms(i,j)+ 13)<=mos(i,j)<=(12-Nms(i,j)+ 24) then yme(i,j)=yms(i,j)+ 200+mos(i,j)- 24;
if (12-Nms(i,j)+ 25)<=mos(i,j)<=(12-Nms(i,j)+ 36) then yme(i,j)=yms(i,j)+ 300+mos(i,j)- 36;
if (12-Nms(i,j)+ 37)<=mos(i,j)<=(12-Nms(i,j)+ 48) then yme(i,j)=yms(i,j)+ 400+mos(i,j)- 48;
if (12-Nms(i,j)+ 49)<=mos(i,j)<=(12-Nms(i,j)+ 60) then yme(i,j)=yms(i,j)+ 500+mos(i,j)- 60;
if (12-Nms(i,j)+ 61)<=mos(i,j)<=(12-Nms(i,j)+ 72) then yme(i,j)=yms(i,j)+ 600+mos(i,j)- 72;
if (12-Nms(i,j)+ 73)<=mos(i,j)<=(12-Nms(i,j)+ 84) then yme(i,j)=yms(i,j)+ 700+mos(i,j)- 84;
if (12-Nms(i,j)+ 85)<=mos(i,j)<=(12-Nms(i,j)+ 96) then yme(i,j)=yms(i,j)+ 800+mos(i,j)- 96;
if (12-Nms(i,j)+ 97)<=mos(i,j)<=(12-Nms(i,j)+108) then yme(i,j)=yms(i,j)+ 900+mos(i,j)-108;
if (12-Nms(i,j)+109)<=mos(i,j)<=(12-Nms(i,j)+120) then yme(i,j)=yms(i,j)+1000+mos(i,j)-120;
if (12-Nms(i,j)+121)<=mos(i,j)<=(12-Nms(i,j)+132) then yme(i,j)=yms(i,j)+1100+mos(i,j)-132;
if yme(i,j)>iym then yme(i,j)=iym;
end;

```

** If estimated weeks are missing then set stop date to December of start year and constrain by DCI;

```

if yms(i,j)>0 and -3<=wks(i,j)<0 then yme(i,j)=(round (yms(i,j),100))+12;
if yme(i,j)>iym then yme(i,j)=iym;

```

** eflag variables:;

```

if (eflag(i,j) not in (2,3,7) or eflag(i,j)=.) then eflag(i,j)=5;
if eflag(i,j) in (2,3,7) then eflag(i,j)=6;

```

end;

*** 6. If the start and stop years are unknown and the estimated weeks are unknown then set start and stop dates to DLI and DCI, respectively.:;

```

if -3<=ys(i,j)<=-1 and -3<=ye(i,j)<=-1 and cwks(i,j)=0 and -3<=wks(i,j)<0 then do;
  yme(i,j)=iym;
  yms(i,j)=dliym;

```

** eflag variables:;

```

if j in (1,5) then eflag(i,j)=6;
else if j in (2,3,4,6,7,8) then eflag(i,j)=7;

```

end;

*** Convert start and stop dates of welfare spells into cms format;

```

csm(i,j)=(round(yms(i,j),100)-198000)*.12+(yms(i,j)-round(yms(i,j),100));
cem(i,j)=(round(yme(i,j),100)-198000)*.12+(yme(i,j)-round(yme(i,j),100));

```

** Added to zero out people who back report;

```
cdli=(round(dliym,100)-198000)*.12+(dliym-round(dliym,100))-1;
```

/** This portion of the program creates a variable that determines who receives afdc in the household. It collapses the answers to 8 categories. The coding is the following -

- 1=respondent only
- 2=spouse/partner only
- 3=child only
- 4=respondent and spouse/partner
- 5=respondent and child
- 6=spouse/partner and child
- 7=respondent and spouse/partner and child
- 8=other

The first 7 categories may include an 'other' person as captured by response categories 4 (other relatives in household) and 5 (other (specify)) in the original question. If only 'other' person is listed as receiving, then the 8th answer category is used in the created variable. ***/

```
if rcvr(i,j)=1 and rcvs(i,j)=0 and rcvc(i,j)=0 then who(i,j)=01;  
if rcvr(i,j)=0 and rcvs(i,j)=1 and rcvc(i,j)=0 then who(i,j)=02;  
if rcvr(i,j)=0 and rcvs(i,j)=0 and rcvc(i,j)=1 then who(i,j)=03;  
if rcvr(i,j)=1 and rcvs(i,j)=1 and rcvc(i,j)=0 then who(i,j)=04;  
if rcvr(i,j)=1 and rcvs(i,j)=0 and rcvc(i,j)=1 then who(i,j)=05;  
if rcvr(i,j)=0 and rcvs(i,j)=1 and rcvc(i,j)=1 then who(i,j)=06;  
if rcvr(i,j)=1 and rcvs(i,j)=1 and rcvc(i,j)=1 then who(i,j)=07;  
if -3<=rcvr(i,j)<0 then who(i,j)=rcvr(i,j);  
if rcvr(i,j)=0 and rcvs(i,j)=0 and rcvc(i,j)=0 and (rcvf(i,j)=1 OR rcvo(i,j)=1) then who(i,j)=08;
```

/** This portion of the program uses the category reported by the respondent to create an estimated amount. The estimated amount is the midpoint rounded down. Note that the 12th category for afdc, fs & other lists \$1251 as the amount. This amount was chosen since the category is unbounded - the number represents one dollar above the lower bound. Similarly the 6th (highest) category for wic is \$101, \$1 above the lower bound, \$100.**/

```
/** afdc **/  
if i=1 then do;  
    inc(i,j)=inc(i,j);  
    if -3<=inc(i,j)<=-1 and 1<=einc(i,j)<=10 then inc(i,j)=(einc(i,j)*100)-50;  
    if -3<=inc(i,j)<=-1 and einc(i,j)=11 then inc(i,j)=1125;  
    if -3<=inc(i,j)<=-1 and einc(i,j)=12 then inc(i,j)=1251;  
end;  
  
/** wic **/  
if i=2 then do;  
    inc(i,j)=inc(i,j);  
    if -3<=inc(i,j)<=-1 and 1<=einc(i,j)<=5 then inc(i,j)=(einc(i,j)*20)-10;  
    if -3<=inc(i,j)<=-1 and einc(i,j)=6 then inc(i,j)=101;  
end;  
  
/** food stamps **/  
if i=3 then do;  
    inc(i,j)=inc(i,j);  
    if -3<=inc(i,j)<=-1 and 1<=einc(i,j)<=10 then inc(i,j)=(einc(i,j)*100)-50;
```

```

if -3<=inc(i,j)<=-1 and einc(i,j)=11 then incc(i,j)=1125;
if -3<=inc(i,j)<=-1 and einc(i,j)=12 then incc(i,j)=1251;
end;

/** other assistance */
if i=4 then do;
incc(i,j)=inc(i,j);
if -3<=inc(i,j)<=-1 and 1<=einc(i,j)<=10 then incc(i,j)=(einc(i,j)*100)-50;
if -3<=inc(i,j)<=-1 and einc(i,j)=11 then incc(i,j)=1125;
if -3<=inc(i,j)<=-1 and einc(i,j)=12 then incc(i,j)=1251;
end;

if incc(i,j)>-4 then aflag(i,j)=0;

END;                                     /** [E3] */
END;                                     /** [E2] */

```

***** HAND EDITS (III) Based on data and questionnaire problems;

** These respondent reported receiving WIC at DLI, but did know whether stopped for more than a month
and so we do not have any information on stop date. We put the start date of the spell as DLI and
stop date as DCI, and put eflag=7, incc=-3 and who=-3.;
if pubid=8147 then do;
csmw11=dlicm; cemw11=doicm; eflagw11=7; aflagw11=0; inccw11=-3; whow11=-3;
end;

** These respondents reported receiving at DLI, currently not receiving and no periods of 1 month or
more when they did not receive and so we take it that they were receiving till less than a month
before the interview.;
if pubid in (274,595,7762) then do;
cema11=DOICM-1; Neflaga11=1;
end;
if pubid in (365,388,441,1275,1988,3811,3919,4212,4357,5459,
5900,6460,6605,6682,6923,7483,8093,8277,8637,8778) then do;
cemw11=DOICM-1; Neflagw11=1;
end;
if pubid in (594,714,1436,4329,4795,5296,5942,7399) then do;
cemf11=DOICM-1; Neflagf11=1;
end;
if pubid in (1953,2749,5682,6013,6422,8374,8939,8969) then do;
cemo11=DOICM-1; Neflago11=1;
end;

** These respondents reported currently receiving, a period of 1 month or more when they did not receive, have a valid stop date but when asked if they received again, said NO so did not go through the second loop. In such cases we create another loop from the month after the reported stop month to DCI, and put eflag=7, incc=-3 and who=-3. However, if the stop date of the first

spell reported is DCI we do not add the second artificial spell.;
if pubid in (1532,6815,6920) then do;
csma12=cema11+1; cema12=doicm; eflaga12=7; aflaga12=0; incca12=-3; whoa12=-3;
end;

if pubid in (392,2210,3542,5367,7880) then do;
csmw12=cemw11+1; cemw12=doicm; eflagw12=7; aflagw12=0; inccw12=-3; whow12=-3;
end;
if pubid=8867 then do;
csmw13=cemw12+1; cemw13=doicm; eflagw13=7; aflagw13=0; inccw13=-3; whow13=-3;
end;
if pubid in (43,3830,5293,5466,6463,8822) then do;
csmf12=cemf11+1; cemf12=doicm; eflagf12=7; aflagf12=0; inccf12=-3; whof12=-3;
end;
if pubid=3058 then do;
csmo12=cemo11+1; cemo12=doicm; eflago12=7; aflago12=0; incco12=-3; whoo12=-3;
end;

** These respondents reported currently receiving and stopping for more than a month but were never asked if they received again. In such cases we create another loop/spell from the month after the reported stop date till DCI and put eflag=7, incc=-3 and who=-3. However if the stop date is DCI, then we do not add the artificial loop/spell.;

if pubid in (1420,4074,5800,5921,6405,6492,6821,6991,8254) then do;
csmw22=cemw21+1; cemw22=doicm; eflagw22=7; aflagw22=0; inccw22=-3; whow22=-3;
end;
if pubid in (2642,4433,6589) then do;
csmf22=cemf21+1; cemf22=doicm; eflagf22=7; aflagf22=0; inccf22=-3; whof22=-3;
end;
if pubid=4060 then do;
csmo22=cemo21+1; cemo22=doicm; eflago22=7; aflago22=0; incco22=-3; whoo22=-3;
end;

** These respondents went through the Never Prior section even though they had reported receiving the program in earlier rounds and so we truncate the start date to the month after the latest stop date reported in a previous round.;

if LAST_A~= and csma21~= then do;
ocsma21=csma21;
if csma21<=LAST_A then do;
csma21=LAST_A+1;
end;
end;
if LAST_W~= and csmw21~= then do;
ocsmw21=csmw21;
if csmw21<=LAST_W then do;
csmw21=LAST_W+1;
end;
end;
if pubid=5921 then do;
csmw21=266; cemw21=277; eflagw21=7; aflagw21=0; inccw21=-3; whow21=-3;
csmw22=.; cemw22=.; eflagw22=.; aflagw22=.; inccw22=.; whow22=.;
end;

```

if pubid=6492 then do;
csmw21=267; cemw21=278; eflagw21=7; aflagw21=0; inccw21=-3; whow21=-3;
csmw22=.; cemw22=.; eflagw22=.; aflagw22=.; inccw22=.; whow22=.;
end;
if pubid=6821 then do;
csmw21=265; cemw21=277; eflagw21=7; aflagw21=0; inccw21=-3; whow21=-3;
csmw22=.; cemw22=.; eflagw22=.; aflagw22=.; inccw22=.; whow22=.;
end;
if LAST_F~= and csmf21~= then do;
ocsmf21=csmf21;
if csmf21<=LAST_F then do;
csmf21=LAST_F+1;
end;
end;
if LAST_O~= and csmo21~= then do;
ocsmo21=csmo21;
if csmo21<=LAST_O then do;
csmo21=LAST_O+1;
end;
end;

```

***** END OF HAND EDITS (III) *****;

*** Compute eflags for those cases that have not already been computed (eflag is not set if the stop date is before DLI and some other hand edits), Neflag is. Here we substitute the Neflag values.;

```

if p1210>-4 then do;
do i=1 to 4;
do j=1 to 8;
if eflag(i,j)=. and Neflag(i,j)~= then do;
eflag(i,j)=Neflag(i,j);
end;
end;
end;
end;

```

/** This part of the program creates the monthly array variables - those that indicate whether the respondent received a particular government program for each month and the monthly amount received. **/

```

C=0;
do l=1 to doicm; /** [B4] ***/
C=C+1;
do j=1 to 8; /** [B5] **/


if csm(1,j)<=C<=cem(1,j) then do;
A(l)=eflag(1,j);
AHHM(l)=who(1,j);
AAMT(l)=incc(1,j);
aodd=0;
if AAMT(l)>1000 then do;

```

```

aodd=AAMT(l);
end;
if A(l)>0 and (AAMT(l)=. or AHHM(l)=.) then do;
ack=ack+1;
end;
if AAMT(l)>0 and (A(l)=0 or AHHM(l)=.) then do;
ackm=ackm+1;
end;
if AHHM(l)>0 and (A(l)=0 or AAMT(l)=.) then do;
ackh=ackh+1;
end;
if AAMT(l)=0 then dummya=1;
end;
if csm(2,j)<=C<=cem(2,j) then do;
W(l)=eflag(2,j);
WHHM(l)=who(2,j);
WAMT(l)=incc(2,j);
wodd=0;
if WAMT(l)>1000 then do;
wodd=WAMT(l);
end;
if W(l)>0 and (WAMT(l)=. or WHHM(l)=.) then do;
wck=wck+1;
end;
if WAMT(l)>0 and (W(l)=0 or WHHM(l)=.) then do;
wckm=wckm+1;
end;
if WHHM(l)>0 and (W(l)=0 or WAMT(l)=.) then do;
wckh=wckh+1;
end;
if WAMT(l)=0 then dummyw=1;
end;
if csm(3,j)<=C<=cem(3,j) then do;
F(l)=eflag(3,j);
FHHM(l)=who(3,j);
FAMT(l)=incc(3,j);
fodd=0;
if FAMT(l)>1000 then do;
fodd=FAMT(l);
end;
if F(l)>0 and (FAMT(l)=. or FHHM(l)=.) then do;
fck=fck+1;
end;
if FAMT(l)>0 and (F(l)=0 or FHHM(l)=.) then do;
fckm=fckm+1;
end;
if FHHM(l)>0 and (F(l)=0 or FAMT(l)=.) then do;
fckh=fckh+1;
end;

```

** This fixes a problem in round 3;

```

if fhdm=. and f(1)>0 and famt(1) ~= . then do;
    fhdm(1)=-3;
    fckneg=1;
end;
if famt(1)=0 then dummyf=1;
end;
if csm(4,j)<=C<=cem(4,j) then do;
O(l)=eflag(4,j);
OHHM(l)=who(4,j);
OAMT(l)=incc(4,j);
    oodd=0;
    if oamt(1)>1000 then do;
        oodd=oamt(1);
    end;
    if o>0 and (oamt=.=. OR ohhm=.=.) then do;
        ock=ock+1;
    end;
    if oamt(1)>0 and (o(1)=0 OR ohhm(1)=.=.) then do;
        ockm=ockm+1;
    end;
    if ohhm(1)>0 and (o(1)=0 OR oamt(1)=.=.) then do;
        ockh=ockh+1;
    end;
    if oamt(1)=0 then dummyo=1;
end;
end;
end;                                /* [e5] */
/* [e4] */

```

** (new r6) If start date before DLI and R went through this section previously then ignore information for months prior to DLI.;

```

C=0;
do l=1 to doicm; /* [B6] */
C=C+1;
do j=1 to 8; /* [B7] */

```

```

if csm(1,j)<=C<=cdli and p010=1 and csm(1,j)~=.=. then do;
A(l)=0;
AHHM(l)=.=.;
AAMT(l)=.=.;
end;
if csm(2,j)<=C<=cdli and p010=1 and csm(2,j)~=.=. then do;
W(l)=0;
WHHM(l)=.=.;
WAMT(l)=.=.;
end;
if csm(3,j)<=C<=cdli and p010=1 and csm(3,j)~=.=. then do;
F(l)=0;
FHFM(l)=.=.;
FAMT(l)=.=.;
end;
if csm(4,j)<=C<=cdli and p010=1 and csm(4,j)~=.=. then do;

```

```

O(l)=0;
OHHM(l)=.;
OAMT(l)=.;
end;

END; /** [E7] */
END; /** [E6] */

```

***** HAND EDITS (IV) Based on data problems;

** When Respondents reported overlapping spells with amount receiving and/or person receiving different for these spells we put -3 for all the months variables as we cannot decide which of the two values to use. Note: Convert all months receiving indicator from 9 to -3 after part C-I.;

```

if pubid=6670 then do;
do l=242 to 254;
W(l)=9; WAMT(l)=-3; WHHM(l)=-3;
end;
end;
if pubid=3811 then do;
do l=265 to 275;
F(l)=9; FAMT(l)=-3; FHFM(l)=-3;
end;
end;
if pubid=8147 then do;
do l=265 to 276;
F(l)=9; FAMT(l)=-3; FHFM(l)=-3;
end;
end;

```

** When respondents reported multiple spells, which overlapped and the dates of one the overlapping spells were invalid and so had to be imputed, we put -3 for all the months variables. This is because we do not know which values to put for the months receiving variables (A,F,O,W) for the overlapping months. Note: Change all months receiving indicator from 9 to -3 after part C-I;

```

if pubid=24 then do;
A276=9; A277=9; AAMT276=-3; AAMT277=-3; AHFM276=-3; AHFM277=-3;
W276=9; W277=9; WAMT276=-3; WAMT277=-3; WHHM276=-3; WHHM277=-3;
do l=266 to 276;
F(l)=9; FAMT(l)=-3; FHFM(l)=-3;
end;
end;
if pubid=531 then do;
do l=252 to 281;
W(l)=9; WAMT(l)=-3; WHHM(l)=-3;
end;
end;
if pubid=825 then do;
do l=36 to 277;
F(l)=9; FAMT(l)=-3; FHFM(l)=-3;
end;
end;
if pubid=2012 then do;

```

```

do l=180 to 276;
O(l)=9; OAMT(l)=-3; OHHM(l)=-3;
end;
end;
if pubid=5328 then do;
do l=216 to 278;
O(l)=9; OAMT(l)=-3; OHHM(l)=-3;
end;
end;
if pubid=5774 then do;
do l=205 to 277;
O(l)=9; OAMT(l)=-3; OHHM(l)=-3;
end;
end;
if pubid=5787 then do;
do l=272 to 277;
A(l)=9; AAMT(l)=-3; AHHM(l)=-3;
end;
end;
if pubid=6489 then do;
do l=217 to 276;
A(l)=9; AAMT(l)=-3; AHHM(l)=-3;
end;
end;
if pubid=7975 then do;
do l=252 to 281;
O(l)=9; OAMT(l)=-3; OHHM(l)=-3;
end;
end;

```

***** END OF HAND EDITS (IV) *****;

END; /** [E1] **/

/**3. This portion of the program creates the yearly variables - those that calculate the number of months in a year R received as well as the yearly amounts received. **/

```

array M      (24) M80-M99 M00 M01 M02 M03;
array AMT    (24) AMT80-AMT99 AMT00 AMT01 AMT02 AMT03;
array na     (24) na80-na99 na00 na01 na02 na03;
array nn     (24) nn80-nn99 nn00 nn01 nn02 nn03;
array allamt (4,288) AAMT001-AAMT288 WAMT001-WAMT288 FAMT001-FAMT288
OAMT001-OAMT288;

```

```

do j=1 to 24;
  if gprg_ever>-4 OR P1210>-4 OR p1210=-5 then do;
    M(j)=0;
    ttlm=0;
    AMT(j)=0;
    na(j)=0;
    nn(j)=0;

```

```

end;
end;

** Calculate total no. months to divide the problematic amts over;

** initialize month counter for correction of problematic amounts reported;

amnths=0;
fmnths=0;
omnths=0;
wmnths=0;

if gprg_ever>-4 OR P1210>-4 OR p1210=-5 then do;
  do L=1 to 288;
    if a(l)>0 and aamt(l)>1000 and aodd>0 then do;
      amnths=amnths+1;
    end;
    if f(l)>0 and famt(l)>1000 and fodd>0 then do;
      fmnths=fmnths+1;
    end;
    if o(l)>0 and oamt(l)>1000 and oodd>0 then do;
      omnths=omnths+1;
    end;
    if w(l)>0 and wamt(l)>1000 and wodd>0 then do;
      wmnths=wmnths+1;
    end;
  end;
end;

if gprg_ever>-4 OR P1210>-4 OR p1210=-5 then do; /* [b8] */

do j=1 to 24; /* [b9] */
  do l=((j-1)*12)+1 to (j*12); /* [b10] */
    if aamt(l)>1000 and amnths>0 and a(l)>0 then do; /*divide the amount by the no. of
months*/
      aamt(l)=ROUND(aamt(l)/amnths, 1);
    end;
    if wamt(l)>1000 and wmnths>0 and w(l)>0 then do;
      wamt(l)=ROUND(wamt(l)/wmnths, 1);
    end;
    if famt(l)>1000 and fmnths>0 and f(l)>0 then do;
      famt(l)=ROUND(famt(l)/fmnths, 1);
    end;
    if oamt(l)>1000 and omnths>0 and o(l)>0 then do;
      oamt(l)=ROUND(oamt(l)/omnths, 1);
    end;
  end;

if M(j)>=0 and (A(l)>0 or W(l)>0 or F(l)>0 or O(l)>0) then do;
  M(j)=M(j)+1;
end;

```

```

if a(l)>1 then M(j)=-3;
if w(l)>1 then M(j)=-3;
if f(l)>1 then M(j)=-3;
if o(l)>1 then M(j)=-3;

do i=1 to 4;
  if -3<=allamt(I,L)<=-1 then do;
    nn(j)=nn(j)+1;
    amt(j)=allamt(I,L);
  end;
  if allamt(I,L)>=0 then do;
    na(j)=na(j)+1;
    if na(j)=1 and nn(j)=0 then amt(j)=allamt(I,L);
    if na(j)>1 and nn(j)=0 then amt(j)=amt(j)+allamt(I,L);
    if M(j)=-3 then amt(j)=-3;
  end;
end;

END;                                     /** [E10] ***/
END;                                     /** [E9] ***/

```

```

do l=1 to 288;
  if TTLM>=0 and (a(l)>0 OR w(l)>0 OR f(l)>0 OR o(l)>0) then do;
    TTLM=TTLM+1;
  end;
  if a(l)>1 then TTLM=-3;
  if w(l)>1 then TTLM=-3;
  if f(l)>1 then TTLM=-3;
  if o(l)>1 then TTLM=-3;
end;

end;                                     /** [e8] ***/

```

totalm=TTLM;

```

***** HAND EDITS (IV') Based on data problems;
if pubid=6670 then do; do l=242 to 254; W(l)=-3; end; end;
if pubid=3811 then do; do l=265 to 275; F(l)=-3; end; end;
if pubid=8147 then do; do l=265 to 276; F(l)=-3; end; end;
if pubid=24 then do; A276=9; A277=9; W276=9; W277=9; end;
if pubid=24 then do; do l=266 to 276; F(l)=-3; end; end;
if pubid=531 then do; do l=252 to 281; W(l)=-3; end; end;
if pubid=825 then do; do l=36 to 277; F(l)=-3; end; end;
if pubid=2012 then do; do l=180 to 276; O(l)=-3; end; end;
if pubid=5328 then do; do l=216 to 278; O(l)=-3; end; end;
if pubid=5774 then do; do l=205 to 277; O(l)=-3; end; end;
if pubid=5787 then do; do l=272 to 277; A(l)=-3; end; end;
if pubid=6489 then do; do l=217 to 276; A(l)=-3; end; end;
if pubid=7975 then do; do l=252 to 281; O(l)=-3; end; end;
***** END OF HAND EDITS (IV') ****;

```

```
/** Create the final variables by combining information from earlier rounds. **/
```

```
do j=1 to 24;
  if gprg_ever=. then gprg_ever=0;
  if gtotal(j)=-4 then gtotal(j)=0;
  if gatotal(j)=-4 then gatotal(j)=0;

  if M(j)>=0 and gtotal(j)>=0 then do;
    gtotal(j)=M(j)+gtotal(j);
  end;
  if M(j)<0 OR gtotal(j)<0 then do;
    gtotal(j)=-3;
  end;
  if AMT>(j)=0 and gatotal(j)>=0 then do;
    gatotal(j)=AMT(j)+gatotal(j);
  end;
  if AMT(j)<0 OR gatotal(j)<0 OR (gtotal(j)<0 and gtotal (j)~=-4) then do;
    gatotal(j)=-3;
  end;
  if AMT(j)=0 and ogatotal(j)=-4 and gtotal(j)=0 then do;
    gatotal(j)=-4;
  end;
end;

if TTLM>=0 and gprg_ever>=0 then do;
  TTLM=TTLM+gprg_ever;
end;
if TTLM<0 OR gprg_ever<0 then do;
  TTLM=-3;
end;
```

```
/** Create the final datasets. **/
```

```
/** Take care of non-interviews and valid skips **/
```

```
do j=1 to 24;
  if p1210=-4 then do;
    gtotal(j)=-4;
    gatotal(j)=-4;
    TTLM=-4;
  end;
  if p1210=-5 then do;
    gtotal(j)=-5;
    gatotal(j)=-5;
    TTLM=-5;
  end;
end;
```

```
** Identify those cases that stopped before DLI.;

if p1210>-4 then do;
  do i=1 to 4;
```

```
do j=1 to 8;
if 0<=cem(i,j)<dlicm then do;
cembfdli(i,j)=1;
cmd=i;
end;
end;
end;
end;

/** Change all invalid skips to -3 **/

array vars TTLM gtotal80-gtotal99 gtotal00 gtotal01 gtotal02 gtotal03
      gatotal80-gatotal99 gatotal00 gatotal01 gatotal02 gatotal03;

do over vars;
  if vars=-2 then vars=-3;
  if vars=-1 then vars=-3;
end;

endsas;
```
